# PharmaSUG China

# An implementation of XSL-FO techniques to convert define.pdf from define.xml

Kyle Chang, PAREXEL International, Taipei, Taiwan

## ABSTRACT

It's getting more common and important for the electronic submissions of clinical data to include define.xml, which specifies the CDISC standard for providing case report tabulation (CRT) data definitions in XML format. Although define.xml has proven to be a useful mechanism can easily help regulatory reviewers to navigate transmission of CRT metadata, it is not able to provide original features (e.g., hyperlinks, bookmarks…etc.) while they're printed out. One solution is to generate a printable define.pdf document with the same content as the define.xml. The printable define.pdf file accommodates bookmarks and hyperlinks functionality for online review, and it can also be printed out for hardcopy review. Providing define.pdf documents is not only to have an easy way to review clinical data submission package, but also to fulfill submission requirements for sponsor and regulatory authority.

This paper provides an approach that using Extensible Stylesheet Language Transformations (XSLT) template converting "define.xml" to formatting objects for easy transformation to "define.pdf".

## INTRODUCTION

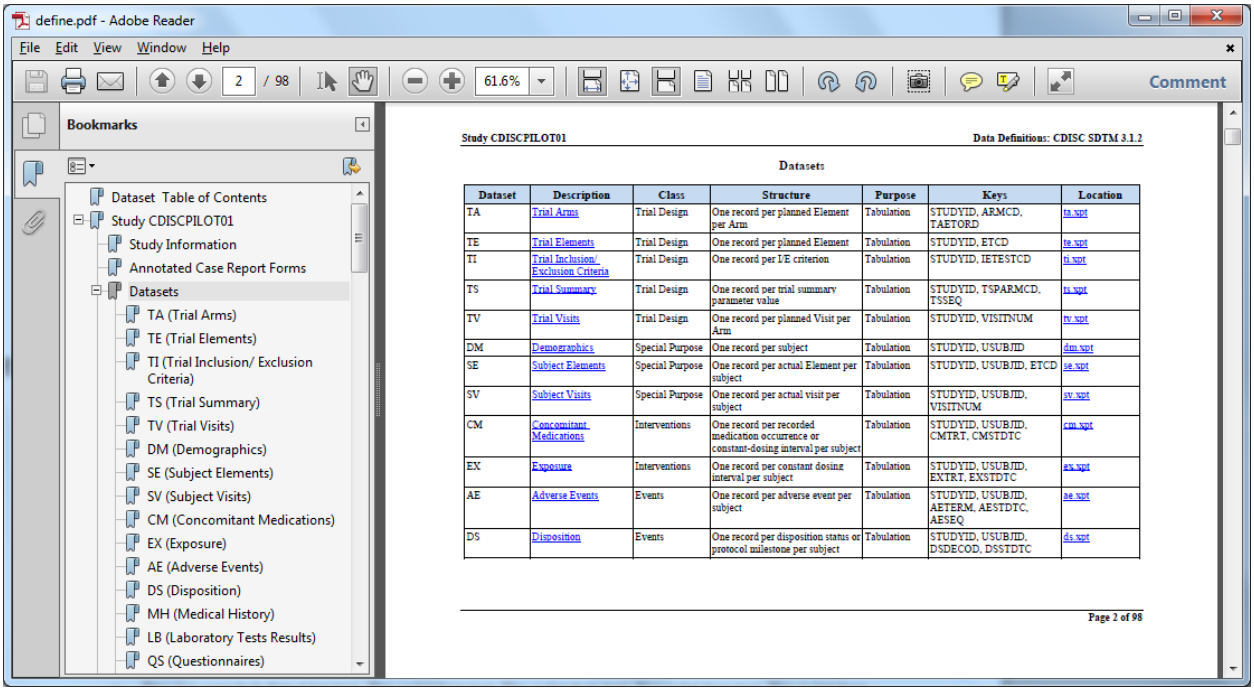
In a regulated industry such as pharmaceutical and biotechnology industry, product submissions for marketing approval currently require define.xml (data definition) files for study data tabulation model (SDTM) , Analysis Data Model (ADaM) data, and even legacy data. Although define.xml files can help the reviewers to navigate submission documents, datasets, and variable definitions, they are not able to be printed out with original content and functions provided in definel.xml. In the industry, every effort is made by sponsors to reduce the review time of data submission. Generating define.pdf file which contains the same contents as the define.xml file may ease the review, hence may reduce the review time.

## THE TYPICAL REQUIREMENTS OF DEFINE.PDF FILE

There are three major expected features of a typical define.pdf file:

1. **Contents**: It should include the same contents as the define.xml, which is viewed through a stylesheet using a web browser. The contents include study information, references to additional documents (e.g., annotated case report form, or study reviewer's guides), dataset information, variable-level metadata, value-level metadata, computational methods, and controlled terminologies (i.e. code lists and external dictionaries).
2. **Table of contents, bookmarks and hyperlinks**: It should contain table of contents, bookmarks and hyperlinks to access information within the document. A table of content should be included at the beginning of the document to facilitate the review. A hierarchical bookmark and hyperlink link to another place in the same document or to external documents (e.g., reference documents or transport files). Bookmarks and hyperlinks provide regulatory reviewers the option to review online.
3. **Page settings**: It should display the define.xml contents within sufficient space of a document. When it is printed out, there are essential elements in the hard copy review, such as header and/or footer information, page numbering, and formatting of the contents within the printable space. These enable the reviewer to print the define.pdf for a hardcopy review, and serve the same purpose as the bookmarks for the electronic version. In addition, all of the table header should repeat at the top of each page for easier review.

All features should also align with the International Conference on Harmonization (ICH) M2 and U.S. FDA PDF specifications. Figure 1 shows an example of define.pdf from CDSIC pilot study documents. In the page layout, the bookmarks are present in left panel to navigate study information, annotated case report forms (aCRFs), supplemental documents, dataset metadata for each specific dataset, variable-level metadata, and specific value-level information. The hyperlinks work in right panel to link information across pages. There are two hyperlinks (in blue underline font) for each dataset. The hyperlink in the description column is connected to the define.pdf page for that dataset. The hyperlink in the location column is connected to the actual dataset outside the define.pdf document. All pages of the document include unique identifying study information to be located in the header, and page number in the footers.




| Dataset | Description | Class | Structure | Purpose | Keys | Location |
|---|---|---|---|---|---|---|
| TA | Trial Arms | Trial Design | One record per planned Element per Arm | Tabulation | STUDYID, ARMCD, TAETORD | ta.xpt |
| TE | Trial Elements | Trial Design | One record per planned Element | Tabulation | STUDYID, ETCD | te.xpt |
| TI | Trial Inclusion/ Exclusion Criteria | Trial Design | One record per I/E criterion | Tabulation | STUDYID, IETESTCD | ti.xpt |
| TS | Trial Summary | Trial Design | One record per trial summary parameter value | Tabulation | STUDYID, TSPARMCD, TSSEQ | ts.xpt |
| TV | Trial Visits | Trial Design | One record per planned Visit per Arm | Tabulation | STUDYID, VISITNUM | tv.xpt |
| DM | Demographics | Special Purpose | One record per subject | Tabulation | STUDYID, USUBJID | dm.xpt |
| SE | Subject Elements | Special Purpose | One record per actual Element per subject | Tabulation | STUDYID, USUBJID, ETCD | se.xpt |
| SV | Subject Visits | Special Purpose | One record per actual visit per subject | Tabulation | STUDYID, USUBJID, VISITNUM | sv.xpt |
| CM | Concomitant Medications | Interventions | One record per recorded medication occurrence or constant-dosing interval per subject | Tabulation | STUDYID, USUBJID, CMTRT, CMSTDTC | cm.xpt |
| EX | Exposure | Interventions | One record per constant dosing interval per subject | Tabulation | STUDYID, USUBJID, EXTRT, EXSTDTC | ex.xpt |
| AE | Adverse Events | Events | One record per adverse event per subject | Tabulation | STUDYID, USUBJID, AETERM, AESTDTC, AESEQ | ae.xpt |
| DS | Disposition | Events | One record per disposition status or protocol milestone per subject | Tabulation | STUDYID, USUBJID, DSDECOD, DSSTDTC | ds.xpt |

**Figure 1. Sample define.pdf from CDISC pilot study**

## RENDER DEFINE.PDF VIA XSL-FO FROM DEFINE.XML

We proposed an approach that read the content of define.xml via XSL and convert to a PDF file. XSL-FO just provides clinical programmers to keep original content and features of define.xml in PDF format. Followings are about our proposed workflow and key ideas of here are a way just. All we need to do is to focus on the layout of printed documents.

### START FROM XSL-FO

XSLT is a language to let you convert XML documents into other XML documents, into HTML documents, or into any other text based documents, or even a PDF file.

The XSL Formatting Objects (XSL-FO) standard is one of the least known parts of the XSLT standard. The XSL-FO can be very useful for such tasks as producing PDF or Postscript files from XML documents. Generally speaking, XSL-FO controls the layout and the presentation of XML documents.

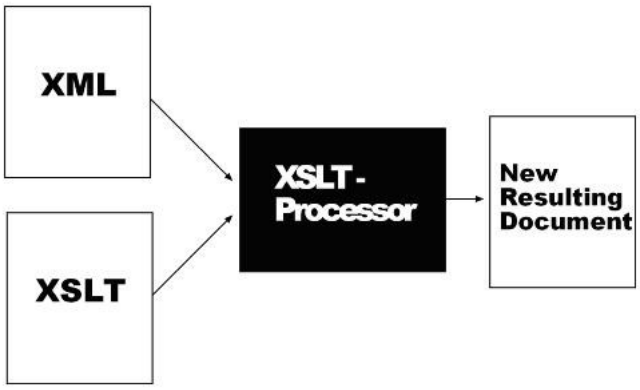



**Figure 2. XSLT Transformation**

To produce a PDF file from the XML file, we need an XSLT stylesheet that converts the XML to XSL-FO. XSL-FO documents are XML files with output information. XSL-FO documents have a structure like this:

```
<?xml version="1.0"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
   <fo:layout-master-set>
        <fo:simple-page-master master-name="A4">
            <!-- Page template goes here -->
        </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-reference="A4">
        <!-- Page content goes here -->
    </fo:page-sequence>
</fo:root>
```

The <fo:root> element is the root element of XSL-FO documents. The root element also declares the namespace for XSL-FO. The <fo:layout-master-set> element contains one or more page templates. Each <fo:simple-page-master> element contains a single page template. Each template must have a unique name (master-name). The master-reference "A4" does not actually describe a predefined page format. It is just a name. You can use any name like "MyPage", "MyTemplate", etc. One or more <fo:page-sequence> elements describe the page contents. We can lay the contents out into pages from define.xml.

### READ METADATA INFORMATION FROM DEFINE.XML

How can a styleshhet read the contents form define.xml? Now, we show a simplest simple XSLT stylesheet that transforms define.xml to a XSL-FO document as follows.

```
<fo:table>
<fo:table-header>
    <fo:table-row>
        <fo:table-cell><fo:block>Dataset</fo:block></fo:table-cell>
        <fo:table-cell><fo:block>Description</fo:block></fo:table-cell>
        <fo:table-cell><fo:block>Class</fo:block></fo:table-cell>
        <fo:table-cell><fo:block>Structure</fo:block></fo:table-cell>
        <fo:table-cell><fo:block>Purpose</fo:block></fo:table-cell>
        <fo:table-cell><fo:block>Keys</fo:block></fo:table-cell>
        <fo:table-cell><fo:block>Location</fo:block></fo:table-cell>
    </fo:table-row>
</fo:table-header>
<fo:table-body>
    <xsl:for-each select="/odm:ODM/odm:Study/odm:MetaDataVersion/odm:ItemGroupDef">
        <fo:table-row>
            <fo:table-cell>
                <fo:block><xsl:value-of select="@Name"/></fo:block>
            </fo:table-cell>
            <fo:table-cell>
                <fo:block>
                    <fo:basic-link>
                        <xsl:attribute name="internal-destination">
                            <xsl:value-of select="@Name"/>
                        </xsl:attribute>
                        <xsl:value-of select="@def:Label"/>
                    </fo:basic-link>
                </fo:block>
            </fo:table-cell>
            <fo:table-cell>
                <fo:block><xsl:value-of select="@def:Class"/></fo:block>
            </fo:table-cell>
            <fo:table-cell>
                <fo:block><xsl:value-of select="@def:Structure"/></fo:block>
            </fo:table-cell>
            <fo:table-cell>
                <fo:block><xsl:value-of select="@Purpose"/></fo:block>
            </fo:table-cell>
            <fo:table-cell>
                <fo:block><xsl:value-of select="@def:DomainKeys"/></fo:block>
            </fo:table-cell>
            <fo:table-cell>
```

```
                <fo:block>
                    <fo:basic-link>
                        <xsl:attribute name="external-destination">
                            <xsl:value-of select="def:leaf/@xlink:href"/>
                        </xsl:attribute>
                        <xsl:value-of select="def:leaf/def:title"/>
                    </fo:basic-link>
                </fo:block>
            </fo:table-cell>
        </fo:table-row>
    </xsl:for-each>
</fo:table-body>
</fo:table>
```

The above stylesheet is formatted/displayed as Figure 1 for a SDTM/ADaM data list. The stylesheet looks long, though, it is only to map the element name of the define.xml to that of XSL formatting object. The layout property defined for the table element is set for the layout property of `<fo:table>` to control the automatic table layout. The XSL `<xsl:for-each>` element allows you to do looping to select every define.xml element of a specified node-set (i.e. `ItemGroupDef`), and the `<xsl:value-of>` element can be used to extract the value of a selected node (i.e. `@Name`, `@Label`, … etc.) from define.xml.

To create the bookmarks and hyperlinks within a PDF document, we can use `<fo:bookmark>` and `<fo:basic-link>` formatting object. They are used to identify an access point (single-target link), by name, and to specify where that access point is within the current document or another external document. A given bookmark may be further subdivided into a sequence of sub-bookmarks to as many levels as our desire. For more details of XSL-FO element, please refer to the World Wide Web Consortium (W3C) website.

### RENDER

Apache™ FOP (Formatting Objects Processor) is a print formatter driven by XSL formatting objects (XSL-FO) and an output independent formatter. It is a Java application that reads a formatting object (FO) tree and renders the resulting pages to a PDF or other specified output (e.g. PS, PCL, AFP… etc.).

Using Apache™ FOP would allow passing parameters to the stylesheet which is something that well-formatted XSL-FO of content layout you have written. The XSLT stylesheets are stored in files with an .xsl extension to make them more accessible to XML editors (save it as `define-pdf.xsl`). Thus, we can type following command with Command Prompt to get a PDF output:

```
> fop -xml define.xml -xsl define-pdf.xsl -pdf define.pdf
```

It reads the generated XSL-FO document and formats it to a PDF document. Now, you've produced your first “define.pdf” with Apache™ FOP! Please open name.pdf in your favorite PDF viewer.

## WHY USE XSL-FO, INSTEAD OF ODS PDF OR ODS RTF?

Compared to XSL-FO, SAS® can also provide other solutions by using ODS utilities (ODS PDF or ODS RTF), they would be more straightforward, but are not able to meet all requirements we mentioned above. We compared their advantages and summarized in below table.

| Features of define.pdf | SAS® ODS PDF | SAS® ODS RTF | XSL-FO |
|---|---|---|---|
| Contents of the define.xml | Yes, but the contents not read from define.xml directly | Yes, but the contents not read from define.xml directly | Yes |
| Hyperlinks with blue underline text | Yes, but the hyperlink is the entire cell of the table and has thicker lines for the cells | Yes, but hyperlink for external documents will only open the first page (or most recently accessed), not the intended page. | Yes |
| Bookmarks | Partial, unable to make the bookmarks hierarchical | Yes | Yes |
| Table of content | Yes | Yes | Yes |
| Header/ footer information | Partial, fail to add long lines under the header for study identifier and above the | Yes, use RTF code in SAS to generate headers/footers and lines, but there is a bug | Yes |

| | | | |
|---|---|---|---|
| | page number in the footer | in SAS v8.2. If the Control Terminology or Comment is too long, it will automatically go to the next page, and the text will be truncated. SAS v9 may fix this bug of ODS RTF | |
| Page number | Yes | Yes | Yes |

## CONCLUSION

A properly functioning define.xml file is an important part of the submission of standardized electronic study datasets and should not be considered optional. In addition to the define.xml, U.S. Center for Drug Evaluation and Research (CDER) prefers a printable define.pdf should be provided if the define.xml cannot be printed properly. There are a lot of tools for define.xml creation, such like SAS© Clinical Standards Toolkit or other third party softwares. But none of them can create a define.pdf very well. Even SAS® ODS PDF and RTF still have some limitations and can't create PDF documents directly. XSL-FO offers powerful properties for controlling the layout and the presentation of XML documents including table of contents, bookmarks, hyperlinks, and proper pagination. One great advantage of using XSL-FO is that it just represents the content as it is in define.xml, and can be a reusable stylesheet for any define.xml following CDISC Define-XML standard, and provides us an efficient tool for electronic submissions.

## REFERENCES

Adams, John H. "Creating a define.xml file for ADaM and SDTM." PharmaSUG 2011.

U.S. Center for Drug Evaluation and Research (CDER). "CDER Common Data Standards Issues Document (Version 1.1)." December 2011. Available at http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM254113.pdf.

U.S. Food and Drug Administration, "Study Data Technical Conformance Guide v1.0 (Public Review)." February 2014. Available at http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM384744.pdf

Clinical Data Interchange Standards Consortium. "Study Data Tabulation Model Metadata Submission Guidelines (SDTM-MSG)." December 2011. Available at http://www.cdisc.org/content3402.

The World Wide Web Consortium (W3C). "Extensible Stylesheet Language (XSL) Version 1.0." October 2001. Available at http://www.w3.org/TR/2001/REC-xsl-20011015/xslspec.html.

Li, Elizabeth; Chesbrough, Carl, "Creating Define.pdf with SAS® Version 9.3 ODS RTF." PharmaSUG 2012. Available at http://www.pharmasug.org/proceedings/2012/AD/PharmaSUG-2012-AD14.pdf

International Conference on Harmonisation, "International Conference on Harmonisation of Technical Requirements For Registration Of Pharmaceuticals for Human Use", ICH M2 EWG Electronic Common Technical Document Specification (Appendix 7), Version 2.6.1., 03-Jun-2008. Available at http://estri.ich.org/eCTD/eCTD_Specification_v3_2_2.pdf

U.S. Food and Drug Administration, "Guidance for Industry - Providing Regulatory Submissions in Electronic Format - Human Pharmaceutical Product Applications and Related Submissions Using the eCTD Specifications", Revision 2, Jun-2008 Available at http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM072349.pdf

U.S. Food and Drug Administration, "Portable Document Format (PDF) Specifications", Version 3.1, 20-Jan-2012. Available at http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM163179.pdf

## ACKNOWLEDGMENTS

My special thanks go to ShengFeng Ho of PAREXEL International, for his review, suggestions, comments, and critiques on this paper.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Kyle Chang
PAREXEL International
22F, Far Glory International Center, No. 200,
Sec. 1, Keelung Road, Taipei, Taiwan 11071, ROC
Kyle.Chang@parexel.com
http://www.parexel.com/