

Adobe XML Architecture

XML Data Package Specification

Version 2.0

Adobe Systems Incorporated

October 2003

© 2003 Adobe Systems Incorporated. All rights reserved.

This publication and the information herein are furnished AS IS, are subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied, or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes, and noninfringement of third-party rights

This page is left intentionally blank.

NOTICE: All information contained herein is the property of Adobe Systems Incorporated.

Any references to company names in the specifications are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe is a registered trademark of Adobe Systems Incorporated in the United States and/or other countries. Unicode is a registered trademark of Unicode, Inc.

All other trademarks are the property of their respective owners.

This limited right of use does not include the right to copy other copyrighted material from Adobe, or the software in any of Adobe's products that use the Portable Document Format, in whole or in part, nor does it include the right to use any Adobe patents, except as may be permitted by an official Adobe Patent Clarification Notice (see the Bibliography).

Adobe XML Data Package has a patent pending in the US. See the official Adobe Patent Clarification Notice (see the Bibliography).

Intellectual Property

Adobe will enforce its intellectual property rights. Adobe's intention is to maintain the integrity of the Adobe XML Architecture standard. This enables the public to distinguish between the Adobe XML Architecture and other interchange formats for electronic documents, transactions and information. However, Adobe desires to promote the use of the Adobe XML Architecture for information interchange among diverse products and applications. Accordingly, Adobe gives anyone permission to use Adobe's intellectual property, subject to the conditions stated below, to:

- Prepare files whose content conforms to the Adobe XML Architecture
- Write drivers and applications that produce output represented in the Adobe XML Architecture
- Write software that accepts input in the form of the Adobe XML Architecture specifications and displays, prints, or otherwise interprets the contents
- Copy Adobe's intellectual property, as well as the example code to the extent necessary to use the Adobe XML Architecture for the purposes above

The condition of such intellectual property usage is:

- Anyone who uses the Adobe intellectual property, as stated above, must include the appropriate intellectual property and patent notices.

This limited right to use the example code in this document does not include the right to use other intellectual property from Adobe, or the software in any of Adobe's products that use the Adobe XML Architecture, in whole or in part, nor does it include the right to use any Adobe patents, except as may be permitted by an official Adobe Patent Clarification Notice (see the Bibliography).

Adobe, the Adobe logo, and Acrobat are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries. Nothing in this document is intended to grant you any right to use these trademarks for any purpose.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA. Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

This page is left intentionally blank.

Table of Contents

- 1 Introduction..... 1
 - 1.1 Background and Goals..... 1
 - 1.2 Notation..... 1
 - 1.3 Terminology..... 1
- 2 The XDP Format..... 3
 - 2.1 XDP Element..... 3
 - 2.2 XDP Namespace..... 3
- 3 XDP Packets..... 5
 - 3.1 Introduction..... 5
 - 3.2 PDF..... 6
 - 3.3 XML Form Data..... 6
 - 3.4 XML Form Template..... 6
 - 3.5 XFA Configuration Information (XCI)..... 6
 - 3.6 XSLT Stylesheet..... 7
 - 3.7 Annotations..... 7
- 4 Bibliography..... 8

This page is left intentionally blank.

1 Introduction

1.1 Background and Goals

This document describes an XML format, known as the "XML Data Package" (XDP), that provides a mechanism for packaging units of PDF content within a surrounding XML [XML] container. The XDP format is intended to be an XML-based companion to PDF. This document describes the XDP format, but does not describe the transformation mechanism between XDP and PDF nor between XDP any other format.

While the PDF format may be most recognized as a visual representation of a document, PDF is actually also a packaging format that encloses many different types and ranges of content. Each of these units of content is referred to as a *subassembly* by this document. For example, a PDF document representing an interactive form may enclose an XML fragment representing the form-data subassembly of the document.

Consider an XML processing application that wishes to process the XML form-data subassembly of a PDF form. Such an application could not directly consume this XML-based subassembly of the PDF because it is enclosed within a non-XML format: PDF.

While extending such an application to interpret the PDF and navigate to the XML data content contained within may be straightforward, this cannot occur by solely employing commonly available XML tools such as an XML parser.

The XDP format provides an alternate means of expressing the PDF document in a manner where the outer packaging is described with an XML-based syntax rather than a PDF-based syntax. Instances of, typically XML, subassemblies are extracted from within the original PDF document and expressed as content within an XDP. After extracting these subassemblies, we are left with the remainder of the PDF document. This remainder of the PDF document is enclosed within the XDP as a region of character-encoded content because of the inability for XML to directly enclose binary content. As a result, the XDP contains all of the information that was formerly enclosed within the PDF, though some of the information may now be expressed in XML. All of the information survives the transformation process. Therefore, a PDF document can be transformed into an XDP and subsequently transformed back into a PDF document without loss of information.

A benefit of the XDP format is that PDF documents can now successfully operate directly within XML workflows because the XDP format provides a means for selectively expressing a PDF document in an XML compatible manner without loss of information. Because the transformations are lossless, document workflows can choose arbitrarily when to process documents in a PDF format vs. when to process the same document in an XML-based format.

In addition to providing a format for expressing one or more subassemblies of a PDF document, the XDP format also has the capability to host arbitrary content. This capability to host arbitrary content is also a feature of PDF. In particular, XDP is an XML-based format with an open content model; the format itself does not prescribe a closed set of content that is allowable, and can therefore be arbitrarily extended.

1.2 Notation

This document includes fragments of XML in both normative and non-normative contexts such as describing the XDP format and illustrating the format with examples respectively. In all cases the XML has been formatted with additional white space to improve legibility, however this white space is non-normative.

1.3 Terminology

The following normative terms may be present in this specification. These terms extend the definitions in [\[RefRFC2119\]](#) in ways based upon similar definitions in ISO/IEC 9945-1:1990 [\[POSIX.1\]](#):

by-default

The term “by-default” is to be interpreted as a value or behavior automatically used in the absence of a choice made by the user.

implementation-defined

The term “implementation-defined” is to be interpreted as delegating to the implementation the definition and documentation of the corresponding requirements for correct processing.

may

The term “may” is to be interpreted as an optional feature or permissible behavior. The feature or behavior is not required by this specification but can legitimately be provided by an implementation. For example, one implementation might choose to include a feature because a particular marketplace requires it or because it enhances the product; another implementation might omit the same feature.

must

The term “must” is to be interpreted as a mandatory requirement on the implementation.

must not

The term “must not” is to be interpreted as a mandatory prohibition on the implementation.

should**recommended**

The terms “should” or “recommended” are to be interpreted as an implementation suggestion, but not a requirement.

There might exist valid reasons for a particular implementation not to implement this behavior, but the full implications ought to be understood and carefully considered before deciding not to implement it.

should not**not recommended**

The terms “should not” or “not recommended” are to be interpreted as an implementation suggestion, but not a requirement.

There might exist valid reasons for a particular implementation to implement this behavior, but the full implications ought to be understood and carefully considered before implementing it.

2 The XDP Format

2.1 XDP Element

The XDP format is comprised of only a single element, known as the XDP element, as follows:

```
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/">
  ...xdp-packet... [0..n]
</xdp:xdp>
```

The XDP element encloses zero or more occurrences of content, each known as an XDP packet, that is described in the section *XDP Packets* below. Because the XDP format itself is comprised only of the XDP element, the functionality and behaviour imparted by an XDP is wholly derived from the content within the XDP. It is the content within the XDP element that is of real significance, not the XDP element itself.

2.2 XDP Namespace

The XDP element must belong to the namespace of `http://ns.adobe.com/xdp/`, which is known as the XDP namespace.

The XDP element should make use of explicitly prefixed namespace notation rather than declaring the XDP namespace as a default namespace. If the XDP element declared the XDP namespace as the default namespace it would have the unfortunate side effect of placing any content that lacks namespace information into the XDP namespace itself.

For example, the following example illustrates the discouraged practice of an XDP that expresses the XDP namespace as the default namespace:

```
<xdp xmlns="http://ns.adobe.com/xdp/">
  <xfa:datasets xmlns:xfa="http://www.xfa.org/schema/xfadata/1.0/">
    <xfa:data>
      <book>
        <ISBN>15536455</ISBN>
        <title>Introduction to XML</title>
        <author>
          <firstname>Charles</firstname>
          <lastname>Porter</lastname>
        </author>
      </book>
    </xfa:data>
  </xfa:datasets>
</xdp>
```

Notice in the above example that the XDP element is not prefixed and declares its namespace via the namespace attribute syntax of `xmlns="http://ns.adobe.com/xdp/"`. The impact of this approach is that any descendant content that does not declare a namespace is at risk of inheriting the XDP namespace. Concretely, in the above example, the result is that the "book" fragment resides in the XDP namespace, which is problematic because such an element is certainly not a valid element of the XDP format, and downstream XML processors intending to interpret this element may no longer recognize the fragment because it has inadvertently been namespaced.

Instead, the following example demonstrates the proper way to declare the XDP namespace:

```
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/">
  <xfa:datasets xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/">
    <xfa:data>
      <book>
        <ISBN>15536455</ISBN>
        <title>Introduction to XML</title>
        <author>
          <firstname>Charles</firstname>
          <lastname>Porter</lastname>
        </author>
      </book>
    </xfa:data>
  </xfa:datasets>
</xdp:xdp>
```

In the above example the namespace declaration on the XDP element does not impact the default namespace and therefore the "book" fragment does not inadvertently inherit the XDP namespace.

3 XDP Packets

3.1 Introduction

The role of an XDP packet is to encapsulate a unit of content, such as a PDF subassembly.

All child elements of the XDP element are considered to be XDP packets. Conversely, an XDP packet must be located as a child element of the XDP element. An XDP packet must not belong to the XDP namespace. The application of the XDP namespace on child elements of the XDP element is reserved for future use.

This section will describe the particular packets supported by Acrobat 6.0. However, the XDP format is also able to enclose packets that are implementation-defined to a particular processing application. Acrobat 6.0 or other processing applications may ignore such packets.

Consider the following example XDP:

```
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/">
  <xfa:datasets xmlns:xfa="http://www.xfa.org/schema/xfadata/1.0/">
    <xfa:data>
      <book>
        <ISBN>15536455</ISBN>
        <title>Introduction to XML</title>
        <author>
          <firstname>Charles</firstname>
          <lastname>Porter</lastname>
        </author>
      </book>
    </xfa:data>
  </xfa:datasets>
  <pdf xmlns="http://ns.adobe.com/xdp/pdf/">
    <document>
      <chunk>
        JVBERi0xLjMKJeTjz9IKNSAwIG9iago8PC9MZW5...
        ZQo+PgpzdHJlYW0KeJylWEtv3DYQvutX8FKgPZj...
        Z/iUBGstoTDg9cfVfPPgcPjJDxUnDH7wt3GCtPv...
      </chunk>
    </document>
  </pdf>
  <my:example xmlns:my="http://www.example.com/">
    <my:message>This packet does not represent a PDF subassembly</my:message>
  </my:example>
</xdp:xdp>
```

The above example XDP encloses three XDP packets:

1. The first packet is represented by the `<xfa:datasets>` element that encloses the XML form-data subassembly of a PDF form.
2. The second packet is represented by the `<pdf>` element that encloses an encoded PDF form minus the XML form-data represented by the first packet.
3. The third packet is represented by the `<my:example>` element that encloses content meaningful to the creator of the XDP but does not represent a subassembly of the PDF form.

3.2 PDF

The PDF packet encloses the remainder of the PDF document that resulted from extracting any subassemblies into the XDP.

XML is a text format, and is not designed to host binary content. PDF files are binary and therefore must be encoded into a text format before they can be enclosed within an XML format such as XDP. The most common method for encoding binary resources into a text format, and the method used by the PDF packet, is *base64 encoding* [RFC2045].

The PDF packet has the following format:

```
<pdf xmlns="http://ns.adobe.com/xdp/pdf/">
  <document>
    <chunk>
      ...base64 encoded PDF content...
    </chunk>
  </document>
</pdf>
```

The <chunk> element must enclose a base64 encoded PDF document.

3.3 XML Form Data

The XML form data packet encloses XML data content that may have originated from an Adobe XML form and/or may be intended to be consumed by an Adobe XML form.

The form-data packet is expressed with the <datasets> and <data> element described by the *XFA-Data Handling Specification* [XFADData]. The *XFA-Data Handling Specification* provides the normative definition of these elements.

```
<xfa:datasets xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/">
  <xfa:data>
    ...XML form-data content...
  </xfa:data>
</xfa:datasets>
```

3.4 XML Form Template

The form template packet encloses the definition of an XML form template.

The form template packet is expressed with the <template> element defined by the *XFA-Template 2.0 Syntax Specification* [XFATemplate]. The *XFA-Template Syntax Specification* provides the normative definition of the <template> element. The following description of the <template> element is therefore non-normative:

```
<xfa:template
  xmlns:xfa="http://www.xfa.org/schema/xfa-template/2.0/">
  ...XML form-template content...
</xfa:template>
```

3.5 XFA Configuration Information (XCI)

The XCI packet encloses the configuration settings intended for consumption by an XFA processing application, such as Acrobat 6.0.

The XCI packet is expressed with the <config> element defined by the *XFA-Configuration Information Specification* [XFAConfig]. The *XFA-Configuration Information Specification* provides the normative definition of the <config> element. The following description of the <config> element is therefore non-normative:

```
<xfa:config xmlns:xfa="http://www.xfa.org/schema/xfa-config/2.0/">
  ...XML configuration content...
</xfa:config>
```

3.6 XSLT Stylesheet

The XSLT packet encloses a single XSLT stylesheet. The XDP format may enclose more than one XSLT packet. Refer to the *W3C XSL Transformations* specification [XSLT] for further information on how to configure XSLT processing.

The XSLT packet is expressed with an appropriately namespaced <stylesheet> element as defined by the W3C "XSL Transformations" specification [XSLT]. The following description of the <stylesheet> element is therefore non-normative:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  ...XSL stylesheet elements...
</xsl:stylesheet>
```

3.7 Annotations

The annotations packet encloses collaboration annotations placed upon a PDF document and is expressed via a subset of the Adobe XFDf format [XFDf].

```
<xfdf xmlns="http://ns.adobe.com/xfdf/" xml:space="preserve">
  <f href="Document.pdf"/>
  <ids original="7A0631678ED475F0898815F0A818CFA1"
    modified="BEF7724317B311718E8675B677EF9B4E"
  />
  <annots>
    <text flags="print,nozoom,norotate" page="0" subject="Note"
      rect="271.850464,690.255371,291.850464,708.255371"
      creationdate="D:20030425095243-07'00'"
      name="apYVRecPEj75sYIwSxME7C" ...
    >
      ...
      <popup .../>
    </text>
  </annots>
</xfdf>
```

4 Bibliography

THIS BIBLIOGRAPHY PROVIDES details on books and documents, from both Adobe Systems and other sources, that are referred to in this specification.

Resources from Adobe Systems Incorporated

All of these resources from Adobe Systems are available on the Adobe Solutions Network (ASN) Developer Program site on the World Wide Web, located at

<http://partners.adobe.com/asn/developer/>

Document version numbers and dates given in this Bibliography are the latest at the time of publication; more recent versions may be found on the Web site.

The ASN can also be contacted as follows:

Adobe Solutions Network
Adobe Systems Incorporated
345 Park Avenue
San Jose, CA 95110-2704

(800) 685-3510 (from North America)
(206) 675-6145 (from other areas)

acrodevsup@adobe.com

[Adobe Patent Clarification Notice]

“*Adobe Patent Clarification Notice*”. Available on the Legal Notices page of the ASN Developer Program Web site.

[XFAConfig]

“*XFA Configuration Information Specification 2.0*”, Adobe Systems Incorporated, October 2003
Available from <http://partners.adobe.com/>.

[XFAData]

“*XFA Configuration Information 2.0*”, Adobe Systems Incorporated, October 2003
[Available from <http://partners.adobe.com/>].

[XFATemplate]

“*XFA Template Syntax Specification 2.0*”, Adobe Systems Incorporated, October 2003
Available from <http://partners.adobe.com/>.

[XFDF]

“*XFDF Specification*”, Adobe Systems Incorporated, October 2003
Available from <http://partners.adobe.com/>.

[PDF1.5]

“*PDF Reference, fourth edition, Adobe Portable Document Format, Version 1.5*”, Adobe Systems Incorporated, 2003
Available from <http://partners.adobe.com/>.

Resources from other sources

[POSIX.1]

“*ISO/IEC 9945-1:1990 Information Technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C Language]*”, Institute of Electrical and Electronics Engineers, Inc, 1990.

[RFC2045]

“*Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*”, N. Freed and N. Borenstein, November 1996.

Available at <http://www.ietf.org/rfc/rfc2045.txt>.

[RFC2119]

“*RFC2119: Key words for use in RFCs to Indicate Requirement Levels*”, S. Bradner, March 1997.

Available at <http://www.ietf.org/rfc/rfc2119.txt>

[XML]

“*Extensible Markup Language (XML) 1.0 (Second Edition)*”, World Wide Web Consortium, October 2000.

Available at <http://www.w3.org/TR/REC-xml>.

[XSLT]

“*XSL Transformations (XSLT)*”, World Wide Web Consortium

Available at <http://www.w3.org/TR/xslt>.