[MS-DSEXPORT]:
Document Set Package Format

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Minor	Updated the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.0	Major	Significantly changed the technical content.
04/11/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	3.0	Major	Significantly changed the technical content.
02/11/2013	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/30/2013	3.1	Minor	Clarified the meaning of the technical content.
11/18/2013	3.1	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction 5
1.1 Glossary 5
1.2 References 5
1.2.1 Normative References 5
1.2.2 Informative References 6
1.3 Structure Overview (Synopsis) 6
1.4 Relationship to Protocols and Other Structures 6
1.5 Applicability Statement 6
1.6 Versioning and Localization 6
1.7 Vendor-Extensible Fields 6

2 Structures 7
2.1 Property Manifest 8
2.1.1 Namespaces 8
2.1.2 Complex Types 8
2.1.2.1 DSProperties 8
2.1.2.1.1 Child Elements 9
2.1.2.2 DSProperty 9
2.1.2.2.1 Child Elements 9
2.2 File Name Mapping 9
2.2.1 Namespaces 9
2.2.2 Complex Types 9
2.2.2.1 FileNameMapping 9
2.2.2.1.1 Child Elements 10
2.2.2.2 OriginalFileName 10
2.2.2.2.1 Attributes 10

3 Structure Examples 11
3.1 Property Manifest 11
3.2 File Name Mapping 11
3.3 Manifest and binary locations 12

4 Security 14
4.1 Security Considerations for Implementers 14
4.2 Index of Security Fields 14

5 Appendix A: Product Behavior 15

6 Change Tracking 16

7 Index 17

1 Introduction

The Document Set Package Format stores the contents of a document set that has been exported from a document library.

Sections 1.7 and 2 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [MS-OFCGLOS]:

content type
content type identifier
document library
field internal name
URL encode

The following terms are specific to this document:

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[ISO/IEC29500-2:2011] ISO/IEC, "Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 2: Open Packaging Conventions", ISO/IEC 29500-2:2011, 2011, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59576

[MS-WSSCAML] Microsoft Corporation, "Collaborative Application Markup Language (CAML) Structure".

[MS-WSSTS] Microsoft Corporation, "Windows SharePoint Services".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1] Thompson, H.S., Beech, D., Maloney, M., Eds., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

1.3 Structure Overview (Synopsis)

This file format stores the contents of a document set outside a **document library**. A document set is a container for managing a collection of documents. Document sets allow the user to perform actions on a collection of documents such as synchronizing metadata across the documents and moving the document set to a different location. Document sets can be packaged to be moved to another location. This file format enables the document set to retain internal relationships and metadata properties. If the document set package is imported or moved to another location, it can be recreated as a document set.

1.4 Relationship to Protocols and Other Structures

The Document Set Package Format structure is implemented following specifications described in [ISO/IEC29500-2:2011]. It uses relationship parts and a content type part to describe the contents of the package.

1.5 Applicability Statement

This file format maintains the metadata properties and relationships of items in a document set so that it can be moved to a different document library and retain its relationships and properties. The internal structure of a document set package file is not used or modified outside the document library that contains the document set. The document set package file is intended as a storage mechanism so that document sets can be archived or moved to another location.

1.6 Versioning and Localization

This file format has no generic localization mechanism. Individual structures in this file format can have attributes that are specific to localization.

1.7 Vendor-Extensible Fields

This file format consists of a collection of mandatory files in a fixed format. Vendors can add their own application-specific files or properties to the package. Those files MUST have names that are unique within the package.

2 Structures

A Document Set Package Format structure is implemented as specified in [ISO/IEC29500-2:2011] and consists of several XML package parts, as well as files of an arbitrary format that are contained in the package. For the document set itself and for each of the files that it contains, a property manifest is required. The property manifest is a list of all of the metadata properties for the document set or file that it describes. The property manifest for the document set MUST be located at /Resources/Properties.xml. The property manifests for individual files in the document set MUST use the following naming convention: /Resources/*FilePath*/*FileName.extension*_Properties.xml<1> where *FilePath* is the full path to the document, where the document set itself is the root and *FileName* is the name of the file to which the property manifest applies and *extension* is the extension of the file. For example, document.docx in the root of the document set would have a property manifest named document.docx_Properties.xml and be placed in the Resources directory, and subfolder1/subfolder2/document.docx would also have a property manifest named document.docx_Properties.xml but be placed in the Resources/subfolder1/subfolder2 directory. Additionally, each subfolder in the document set, will also have a property manifest located in /FolderProps/*FolderPath*/_Properties.xml where *FolderPath* is the full path to the folder starting from the root of the document set<2>; for example a subfolder named subfolder2 with path subfolder1/subfolder2/ would have a manifest file /FolderProps/Subfolder1/subfolder2/_Properties.xml.

All file names including path inside the document set package MUST be **URL-encoded**. If a file in the document set has a file name which, when URL-encoded, has length greater than or equal to 200 characters, the file name MUST be shortened to a new file name for use within the document set package. The new file name MUST be unique within the document set package and MUST be URL-encoded. The original file extension SHOULD<3> be retained. A mapping of the new file name to the original file name MUST be stored in the package part /Resources/FileNameMapping.xml. If there are multiple mappings, they are all kept in that package part. If there are no mappings, then that package part is not required.

File and package part relationships are specified in [ISO/IEC29500-2:2011], section 9.3. A document set package MUST contain the following relationship types:

- **http://microsoft.com/docset/MainProperties**: Main property manifest package part, which describes the document set properties.
- **http://microsoft.com/docset/FileProperties**: Property manifest package parts.

A document set package MUST contain the following relationship type if any file name is shortened to a new file name because the URL-encoded file name is greater than or equal to 200 characters:

- **http://microsoft.com/docset/FileNameMapping**: Package part that contains file name mappings.

A document set package MUST contain the following relationship type if it contains files other than the parts required for the document set package:

- **http://microsoft.com/docset/File**: Files that are contained in the document set.

A document set package MUST contain the following relationship type if it contains subfolders:

- **http://microsoft.com/docset/Folder**: Subfolders that are contained in the document set.

Also as specified in [ISO/IEC29500-2:2011], the content type of each package part and file MUST be defined in /[Content_Types].xml. A document set package can have but does not require all of the following content types be present. No other content types are allowed.

- **text/xml**: XML package parts
- **application/vnd.openxmlformats-package.relationships+xml**: Relationship parts
- **application/octet-stream**: Files

2.1 Property Manifest

A property manifest is a list of metadata properties and their values. One property manifest describes the properties of the document set itself, and additional property manifests describe the properties of the files and folders within the document set. A separate property manifest is required for each file and folder within the document set. Thus, if a document set contains two files and one subfolder, there are four property manifests: one for the document set, two for the files (one for each file), and one for the folder. Each property manifest for a file or the document set MUST be located at /Resources/. Each property manifest for a folder MUST be located at /FolderProps/

The root element of the property manifest is defined as follows:

```
<xs:element name="Properties" type="DSProperties" />
```

2.1.1 Namespaces

This specification defines and references XML namespaces using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of a specific XML namespace prefix is implementation-specific and not significant for interoperability.

The following table described these namespaces.

Prefix	Namespace URI	Reference
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
(none)	urn:deployment-manifest-schema	

2.1.2 Complex Types

2.1.2.1 DSProperties

The **DSProperties** complex type specifies the **content type** name, **content type identifier**, and properties of the document set or file. This type is defined as follows:

```
<xs:complexType name="DSProperties">
   <xs:sequence>
     <xs:element name="ContentType" type="xs:string" />
     <xs:element name="ContentTypeName" type="xs:string" />
     <xs:element maxOccurs="unbounded" name="Property" type="DSProperty">
     </xs:element>
   </xs:sequence>
 </xs:complexType>
```

2.1.2.1.1 Child Elements

ContentType: The content type identifier of the document set or file. It MUST be formatted as specified in [MS-WSSCAML] section 2.3.1.4.

ContentTypeName: The name of the content type of the document set or file, as specified in [MS-WSSCAML] section 2.4.1.

Property: A property of the document set or file, as specified in [MS-WSSTS] section 2.1.2.9. Lookup field values are ignored on import so it does not matter if they are exported.

2.1.2.2 DSProperty

The **DSProperty** complex type specifies the name and value of a property. This type is defined as follows:

```
<xs:complexType name="DSProperty">
  <xs:sequence>
    <xs:element name="Name" type="xs:string" />
    <xs:element name="Value" type="xs:string" />
    <xs:element name="Type" type="xs:string" />
  </xs:sequence>
</xs:complexType>
```

2.1.2.2.1 Child Elements

Name: The **field internal name** of the property.

Value: The value of the property, which is a value appropriate for the Type as specified in [MS-WSSTS] section 2.3.

Type: The type of the property, as specified in [MS-WSSTS] section 2.1.2.9.1.

2.2 File Name Mapping

The file name mapping XML document specifies the list of documents from the document set whose names within the document set package have been shortened so as not to exceed the 200 character limit when encoded. For each such document, the file name mapping XML document contains an entry that maps the name of the document in the document set package to the original name of the document in the document set.

The root element of the file name mapping is defined as follows:

```
<xs:element name="Files" type="FileNameMapping" />
```

2.2.1 Namespaces

File name mapping does not require a namespace.

2.2.2 Complex Types

2.2.2.1 FileNameMapping

The **FileNameMapping** complex type specifies the mapping from shortened file names to full, original file names, as follows.

```
<xs:complexType name="FileNameMapping">
   <xs:sequence>
     <xs:any minOccurs="0" maxOccurs="unbounded">
     </xs:any>
   </xs:sequence>
 </xs:complexType>
```

2.2.2.1.1 Child Elements

The **xs:any** part specifies a single mapping from a shortened file name to the original file name. The name of each **xs:any** part MUST be the shortened file name and the type of the **xs:any** part MUST be **originalFileName**.

2.2.2.2 OriginalFileName

The **OriginalFileName** complex type specifies the original file name. This type is defined as follows:

```
<xs:complexType name="originalFileName">
  <xs:attribute name="originalFileName" type="xs:string" use="required" />
</xs:complexType>
```

2.2.2.2.1 Attributes

OriginalFileName: The original name of the file.

3 Structure Examples

3.1 Property Manifest

The following is an example of a property manifest with content type "Document Set A" and three properties.

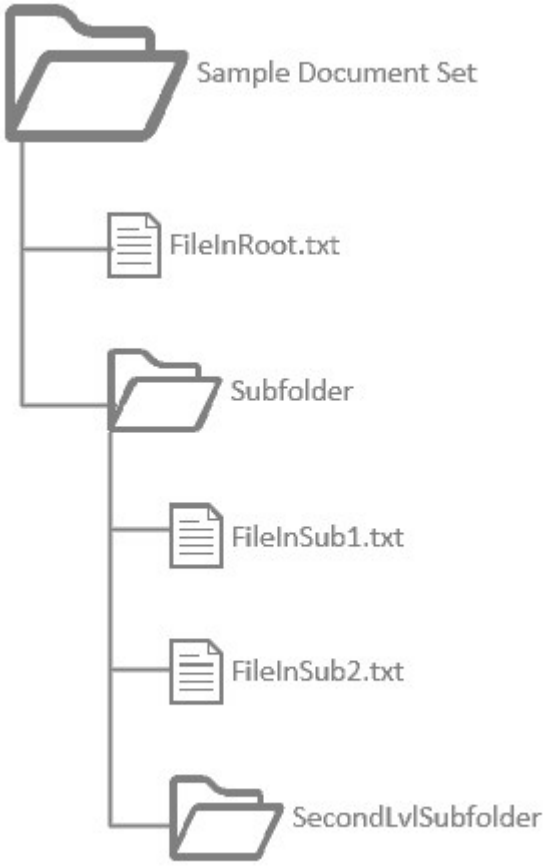
```
<Properties>
  <ContentType>0x0120D5200079D38D8510120240BF8C36A3DFF2A81C</ContentType>
  <ContentTypeName>Document Set A</ContentTypeName>
  <Property>
    <Name>ContentTypeId</Name>
    <Value>0x0120D5200079D38D8510120240BF8C36A3DFF2A81C</Value>
    <Type>ContentTypeId</Type>
  </Property>
  <Property>
    <Name>_ModerationComments</Name>
    <Value></Value>
    <Type>Note</Type>
  </Property>
  <Property>
    <Name>FileLeafRef</Name>
    <Value>Example</Value>
    <Type>File</Type>
  </Property>
</Properties>
```

3.2 File Name Mapping

The following is an example of a file name mapping with two mappings.

```
<Files>
  <__File__1792867176.docx originalFileName="examplefilename.docx" />
  <__File__1940293843.docx originalFileName="examplefilename2.docx" />
</Files>
```

3.3 Manifest and binary locations



For the shown document set structure, the resulting export file would contain the manifests and binaries in the following locations:

```
\FileInRoot.txt – File Binary

\_rels\.rels – Relationships file describing contents of package

\Subfolder\FileInSub1.txt – File Binary

\Subfolder\FileInSub2.txt – File Binary

\Resources\Properties.xml – Document set properties manifest

\Resources\FileInRoot.txt_Properties.xml – File properties manifest

\Resources\Subfolder\FileInSub1.txt_Properties.xml – File properties manifest

\Resources\Subfolder\FileInSub2.txt_Properties.xml – File properties manifest

\FolderProps\Subfolder\_Properties.xml – Folder properties manifest
```

\FolderProps\Subfolder\SecondLvlSubfolder_Properties.xml – Folder properties manifest

The property manifest sample in section 3.1 applies. The content of the relationship file is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
  <Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Type="http://microsoft.com/docset/MainProperties"
Target="/Resources/Properties.xml" Id="R92c64673f7d14941" />
    <Relationship Type="http://microsoft.com/docset/File" Target="/FileInRoot.txt"
Id="Rc736d2b78b03447c" />
    <Relationship Type="http://microsoft.com/docset/Folder"
Target="/FolderProps/Subfolder/_Properties.xml" Id="R85675af82f594244" />
    <Relationship Type="http://microsoft.com/docset/File" Target="/Subfolder/FileInSub1.txt"
Id="Rd8106a36961f4846" />
    <Relationship Type="http://microsoft.com/docset/File" Target="/Subfolder/FileInSub2.txt"
Id="R527cebd070dd4b06" />
    <Relationship Type="http://microsoft.com/docset/Folder"
Target="/FolderProps/Subfolder/SecondLvlSubfolder/_Properties.xml" Id="R7bbfda245f624285" />
```

4 Security

4.1 Security Considerations for Implementers

None.

4.2 Index of Security Fields

None.

5 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2: *FilePath* MUST be empty for SharePoint Server 2010.

<2> Section 2: Folders are not allowed inside document sets in SharePoint Server 2010, so no property files for folders will be contained within the package.

<3> Section 2: SharePoint Server 2010 does not retain the original file extension.

6 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

7 Index

A

Applicability 6

C

Change tracking 16
Common data types and fields (section 2 7, section 2 7)
Complex type
  DSProperties 8
  DSProperty 9
  FileNameMapping 9
  OriginalFileName 10

D

Data types and fields - common (section 2 7, section 2 7)
Details
  common data types and fields (section 2 7, section 2 7)
  DSProperties complex type 8
  DSProperty complex type 9
  file name mapping 9
  FileNameMapping complex type 9
  OriginalFileName complex type 10
  property manifest namespaces 8
  property manifest structure 8
DSProperties complex type 8
DSProperty complex type 9

E

Examples
  File Name Mapping 11
  Manifest and binary locations 12
  Property Manifest 11

F

Fields - security index 14
Fields - vendor-extensible 6
File name mapping 9
File Name Mapping example 11
FileNameMapping complex type 9

G

Glossary 5

I

Implementer - security considerations 14
Index of security fields 14
Informative references 6
Introduction 5

L

Localization 6

M

Manifest and binary locations example 12

N

Namespaces - property manifest 8
Normative references 5

O

OriginalFileName complex type 10
Overview (synopsis) 6

P

Product behavior 15
Property Manifest example 11
Property manifest namespaces 8
Property manifest structure 8

R

References 5
  informative 6
  normative 5
Relationship to protocols and other structures 6

S

Security
  field index 14
  implementer considerations 14
Structures
  DSProperties complex type 8
  DSProperty complex type 9
  file name mapping 9
  FileNameMapping complex type 9
  OriginalFileName complex type 10
  overview (section 2 7, section 2 7)
  property manifest 8
  property manifest namespaces 8

T

Tracking changes 16

V

Vendor-extensible fields 6
Versioning 6