Electronic Document eXchange (EDX) 3.1

Standard Definition

Rev B
1/20/2011

Table Of Contents

Table Of Revisions

Rev.	Date	Author	History
A	10/11/2005	Mike Suleski	Release v3.0
B	1/20/2011	Mike Suleski	Preliminary Release v3.1 – Added ExtDocData.xml file

1. PURPOSE

This document defines the standard for the Electronic Document eXchange (EDX) file format.

2. SCOPE

This document defines the standards for the EDX Transport Package (a file) and its contents (also files). It does not deal with the software required to be compliant with the file standards.

3. REFERENCES

Zip File Format: File http://www.pkware.com/business_and_developers/developer/appnote/ by PKWARE, 1/25/93.

4. DEFINITIONS

EDX	Electronic Document eXchange. A method of transporting documents, batches and associated data from one Document Management System to another, or from a generating program to a Document Management System.
XML	Extensible Markup Language. A standard file format (ref. W3C) for expressing data.

5. OVERVIEW

Electronic and scanned documents and batches tend to be rather easy to create, and moderately easy to store. There are many scanner front end systems that can create enormous numbers of scanned TIFF files and there are many Document Management Systems that permit the storing of these documents. These systems can generate documents and export them from their document stores in addition to providing storage and indexing of the documents.

EDX was created to allow all of these sources of documents to send them to any destination in an efficient, standardized and open manner. This EDX standard document specifies the format of the files required to implement EDX. It does not specify the software or platforms on which EDX will be implemented because it is an open standard. It does not specify the content of the documents being exchanged because the standard is independent of the data.

EDX documents are encapsulated into a single file for the purposes of transport. The Transport *Package* can be moved from one system manually, via a messaging protocol, internally via a network or by any other means of moving a file from one location to another. EDX is independent of the transport mechanism.

Normally, documents transported from one Document Management System to another lose all associated information (indexing, document type, ownership, etc) because the two systems are dissimilar. Even if the two systems are running the same Document Management System software, they will most likely be configured differently. EDX allows for the complete set of associated information to be transported along with the files representing the documents. Further, it permits the mapping of dissimilar configuration information to allow two very different Document Management Systems to maintain all data during the transport process. This mapping is accomplished using Dictionary and Mapping files which are stored in the EDX packages. The Dictionary files describe the data elements specified in the document metadata and the Mapping files define where the data is located in the metadata data records.

6. THE EDX TRANSPORT PACKAGE

6.1 The EDX File Format

The EDX Transport package is contained in a compound file with the extension "EDX". The PKWARE PKZip standard is used to:

1. encapsulate all files within one compound package,
2. reduce the size (compress) for file types that are not already compressed, and
3. optionally encrypt the data.

If encryption is used, then the standard ZIP password protected encryption will be the method. The passwords will, of course, be exchanged outside of the domain of the EDX file. This exchange mechanism is not covered by this document.

6.2 The EDX File Contents

The contents of the EDX file are:

1. The Document eXchange metadata file (MANIFEST.DXD),
2. the data dictionary file (DATADICT.XML),
3. the mapping file (MAPPING.XML),
4. optionally, the extended document data file (EXTDOCDATA.XML),
5. optionally, the document files themselves, and
6. optionally, the archive comment record.

The data files MANIFEST.DXD, DATADICT.XML and MAPPING.XML are stored in the only sub directory in the ZIP file which is named \METADATA. All other files (document files) are stored in the root directory.

6.2.1 The MANIFEST.DXD File

The Document eXchange metadata file (MANIFEST.DXD) is required to exist within the EDX file in order for the EDX file to be a valid EDX transport package (the DXD file can also exist outside an EDX transport package). It specifies control information in its header and contains a record for each folder, document, or batch being created.

The DXD file is an ANSI text file. Each record, with the exception of header and comment records, are comprised of comma separated values. The content of these records is described below.

6.2.2 The DATADICT.XML File

The data dictionary file is required to describe the data elements in the document metadata records (DXD file). The data dictionary file is in XML format and is described in detail below.

6.2.3 The MAPPING.XML File

The mapping file is required to specify where to find each data element in the document metadata records (DXD file). The mapping file is in XML format and is described in detail below.

6.2.4 The EXTDOCDATA.XML File

The extended document data file is to be used to store extra data for documents that may be required by specific applications. The format for this file is defined in this specification but definition of application specific data elements is beyond the scope of the EDX standard.

6.2.5 The Document Files

Each document or batch has exactly one file associated with it. That file may or may not exist within the EDX Transport Package. If it is not contained within the package, then the file will be referred to externally. See below for details on file references.

If contained within the package the file may be directly stored without compression or may be compressed. For file types that already support compression it is more efficient to store the files without attempting further compression. Other files may yield significant space reduction after compressing into the ZIP format. The EDX 3.0 standard makes no requirement of the compression of the files within the ZIP package.

6.2.6 The Archive Comment Record

The PKZip standard specifies an optional comment that may be embedded into each ZIP file. EDX will use this archive comment record to specify the organization that created the EDX transport package and the name of the job (the collection of parameters) under which the package was created. This organization/job pair allows encryption passwords to be distributed outside the domain of the EDX transport package and yet be clear to the receiver of the package which decryption password to apply. Since the archive comment record for the ZIP file is not encrypted, the organization/job pair can be retrieved from the package and the appropriate password for that pair used to decrypt the contents of the package. The format for the archive comment record is:

Org: <organization name>, Job: <job name>

where case is not sensitive in the "Org:" and "Job:" tags and white space after the colon and before the name as well as surrounding the comma is ignored.

The organization name and job name are used to coordinate encryption passwords strictly between the sending and receiving sites. There are no other restrictions on the use of these strings, other than that the organization name cannot contain the character sequence "Job:". The use of the archive comment record, as well as encryption is optional. It is permissible for encryption to be used without the archive comment record providing password hinting. In that case coordinating the password information must be done through some other mechanism, either through software or manual means. That coordination is outside the scope of this document.

7. THE METADATA FILE

7.1 File Format

The Document eXchange Data file is named MANIFEST.DXD. It is an ANSI text file containing

1. header records,
2. comment records and
3. data records.

The following rules apply to the records.

- All records end with a newline.
- Header and comment records begin with a colon (":").
- Data records contain comma delimited fields. All strings (but not character based specifiers) must be enclosed in double quotes.
- Blank lines are ignored

7.2 Header Record

The general form of the Header record is this:

:!EDX <type> <data>

Type	Data	Required	Usage
Version	2.0	Y	Identifies the version of EDX that the file conforms to
Software	Application name	N	Identifies the name of the application that created the file
Creator	User name	N	Identifies the person operating the generating software
Organization	Organization name	N	Identifies the organization/company that created the file
Contact	Contact information	N	Information used to contact the creator in case of problems
Job	Job description	N	Description of the file and its contents
Timestamp	YYYYMMDDhhmmss	N	File creation date and time
Option	F	N	Forces folder creation if it doesn't exist

7.3 The Comment Record

The comment record is ignored by the parsing agent and is used to add human readable information to the file.
The general form of the Comment record is this:
: <comment>

7.4 The Data Record

The data records define the objects to be created by the file and is composed as follows:
<record specifier>,<security specifiers>,<security data>,<routing specifier>,<routing data>,<record data>

7.4.1 The Record Specifier

The record specifier is a single character which defines the type of object that is being created in the record. The record data, which is the variable part of a data record is defined by the specifier.

Record Specifier	Object
F	Folder
D	Document
B	Batch

7.4.2 The Security Specifiers

Security specifiers are a group of zero or more characters. More than one specifier may exist, but only one requiring data may. Thus U and G are mutually exclusive. If no security specifier exists the security data is ignored and the security defaults to none (public).

Specifier	Meaning	Data
	No security (public)	None
I	Internet secure	None
U	Owned by user	User name
G	Owned by group	Group name

7.4.3 The Routing Specifier

The routing specifier is a single optional character. They can be concatenated, but only one may have data associated with it. If no security specifier exists the security data is ignored and the security defaults to none (public).

Specifier	Meaning	Data
	No routing	None
U	Route to user	User name

G	Route to (task) group	Name of task group
L	Route to (distribution) list	Name of distribution list
P	Route to a manual or automated workflow process	Name of process

7.4.4 The Folder Record Data

<drawer>,<folder description>,<folder index fields>,<folder data fields>
Optional fields are left blank, but the comma delimiter must still exist.

Field	Required	Usage
Drawer	Y	Name of the drawer (definition of folder and document fields).
Folder description	N	Non-unique textual identifier of the folder.
Folder index fields	Y	10 values used as indexing information, separated by comma delimiters. At least one required. The rest required as specified by the drawer definition.
Folder data fields	N	10 values used as additional non-indexed information for the folder. Only required if specified as such by the drawer definition.

7.4.5 The Document Record Data

<folder record data>,<document type>,<document description>,<document data fields>,<file specifier>,<file description>
Optional fields are left blank, but the comma delimiter must still exist.

Field	Required	Usage
Folder record data	Y	As specified above. Only required folder index fields need be specified. If the folder already exists, the folder data fields are not required.
Document type	Y	Name of the category that the document belongs to.
Document description	N	Non-unique textual identifier of the document.
Document data fields	N	10 values used as non-indexed information for the document. Only required if specified as such by the drawer definition component of the folder record data.
File Specifier	N	See below.
File Description	Y	See below.

7.4.6 The Batch Record Data

<batch identifier>,<batch description>,<file specifier>,<file description>
Optional fields are left blank, but the comma delimiter must still exist.

Field	Required	Usage
Batch identifier	N	Unique operator assigned identification of the batch.
Batch description	N	Non-unique textual identifier of the batch.
File Specifier	N	See below.
File Description	Y	See below.

7.4.7 The File Specifier

There is always a one-to-one correspondence between files and documents or batches (but not necessarily files *in* the EDX Transport Package). Documents or batches that are multiple pages are contained in multi-page TIFF files.

Specifier	Description
None	Pathless file name that is part of the package; to be placed in the document store.
S	Complete UNC/LFN path and file name of (reference to) a file that corresponds to the document. The file can either remain as an external reference or be copied into the document store.
U	Complete URL of (reference to) a file that corresponds to the document. The file can either remain as an external reference or be copied into the document store.
D	Complete ODMA Document ID of (reference to) a file that corresponds to the document. The file can either remain as an external reference or be copied into the document store.

7.4.8 A Sample File with Comments

```
:!EDX Version 2.0
:!EDX Software PaperClip32 Batch Export
:!EDX Creator Rudolph Peony
:!EDX Organization Acme Industries Inc.
:!EDX Contact Pager:(201)555-1212 Email:rpeony@acmei.com
:!EDX Job Daily Exception Dump
:!EDX Timestamp 19980322124207
:!EDX Option F

: create a folder in drawer "Exceptions"
: it is defined as internet secure and owned by the task group "Reviewers"
: once created it will be routed to a process call "Outliers"
F,IG,"Reviewers",P,"Outliers","Exceptions","Franklin Staples","987-65-4321","Staples","Franklin",,,,,,,"Overdue",,,,,,,,

: now create a few documents to go into that folder
: the ownership will be inherited by the folder just created
: the routed folder will implicitly route the documents
: the only required folder index field is the social security number
: the four documents are:
:       1) a true copy of the original invoice, created with PaperClip PTA, sent with the EDX Transport Package
:       2) a reference to the spreadsheet containing the original order, to be copied into the document store for speed
:       3) a reference to the customer's profile, to be available but not retrieved until the document is opened
:       4) a reference to a form letter to be sent out, the forms being all stored on the company's ftp site
D,,,,,"Exceptions",,"987-65-4321",,,,,,,,,,,,,,,,,"Invoice","Order #3234","Not payed",,,,,,,,,,"230003.tif"
D,,,,,"Exceptions",,"987-65-4321",,,,,,,,,,,,,,,,,"Order","Order #3234","Not payed",,,,,,,,,S,"\\acct.acmei.com\ar\3234.xls"
D,,,,,"Exceptions",,"987-65-4321",,,,,,,,,,,,,,,,,"Profile","Order #3234","Not payed",,,,,,,,,R,"\\acct.acmei.com\cust\987-65-4321.doc"
D,,,,,"Exceptions",,"987-65-4321",,,,,,,,,,,,,,,,,"Letter","Order #3234","Not payed",,,,,,,,,U,"ftp://ftp.acmei.com/forms/overdue.frm"

: finally, a 5 page scanned batch gets associated with the EDX Transport Package
: the TIFF file is a multi-page file
B,,,U,"Harold","1998-03-22/RP01","Harold, please file this where you think is appropriate",,"57.tif"
```

8. DICTIONARY DEFINITION

A dictionary consists of an XML file named DATADICT.XML that contains a set of terms and synonyms. A single file may contain multiple dictionaries. Dictionaries can be ether globally defined or private. Global dictionaries are typically shared among document exchange parties participating in a common marketplace. The use of global dictionaries greatly simplifies the data interchange process. How global dictionaries are defined and administered is beyond the scope of this specification.

8.1 Format

This section describes the structure of the data dictionary file.

8.1.1 <Dictionaries> Element

This element is the root of the document, and may contain multiple <*Dictionary*> elements.

8.1.2 Attributes

None

8.1.3 Nested Elements

This element will contain one or more <Dictionary> elements.

8.1.4 <Dictionary> Element

This element will contain a number of <Term> and <Synonym> elements.

```
<Dictionary Type="Global" Name="PCLIP" Version="1.0">
        <Term></Term>
        <Synonym></Synonym>
</Dictionary>
```

8.1.4.1 Attributes

This element has the following attributes:

1) **Type (Required)** : This attribute contains the type of dictionary defined. The possible values are : *Global* or *Subscriber*. Global dictionaries are those defined and maintained by a document exchange community (marketplace) and they contain the standard set of terms to be used by all subscribers. Subscriber dictionaries will contain terms defined by each client.
2) **Name (Required) :** Name of the dictionary.
3) **Version (Required) :** Version of the defined dictionary.

8.1.4.2 Nested Elements

This element will contain one or more <Term> elements and zero or more <Synonym> elements.

8.1.5 <Term> Element

This element defines each dictionary term.

```
<Term Name="Field" TermType="Field" DataType="string" Length="20"
Required="false" />
```

8.1.5.1 Attributes

This element has the following attributes:

1) **Name (Required) :** This attribute contains name of the term being defined. It is a string value without any spaces.
2) **TermType (Required) :** This attribute describes the type of term being defined. It may contain one of the following values:
    a. ***Field :*** Indicates that the term is a data field
    b. ***DocProp:*** Indicates that term is a document property
    c. ***PkgProp:*** Indicates that term is a package property
3) **DataType (Required) :** This attribute describes the data type of the defined term. It may contain one of the following values:
    a. ***string:*** character data
    b. ***boolean:*** Boolean (true/false) data
    c. ***dateTime:*** Date Time data
    d. ***date:*** Date only
    e. ***time:*** Time only
    f. ***number:*** Numeric data
4) **Length (Required) :** This attribute contains the length of a string DataType
5) **Format (Optional) :** Contains format of the field in the package according to the defined DataType
6) **DecimalDigits (Optional) :** Contains number of decimal places on number data.
7) **Required (Optional) :** Used on terms of TermType DocProp and PkgProp to indicate whether the entry is required. Possible values are: *true* or *false*.

8.1.5.2 Nested Elements

None

8.1.6 <Synonym> Element

This element defines each Synonym in the dictionary.

```
<Synonym Name="FirstName" Term="FName" />
```

8.1.6.1 Attributes

This element has the following attributes:

1) **Name (Required) :** This attribute contains the name of the synonym being defined. It is a string value without any spaces.
2) **Term (Required) :** This attribute contains the value of the defined Term that this synonym is equal to.

8.1.6.2 Nested Elements

None

8.2 Sample file

The following is a sample of how a dictionary file would look like.

```
<?xml version="1.0"?>
<Dictionaries>
	<Dictionary Type="Global" Name="CommonTerms" Version="1.0">
		<Term Name="FName" TermType="Field" DataType="string" Length="20" />
		<Term Name="LName" TermType="Field" DataType="string" Length="20" />
		<Term Name="DocumentType" TermType="DocProp" DataType="string"
Length="100" Required="true" />
		<Term Name="PackageVersion" TermType="PkgProp" DataType="string"
Length="10" Required="true" />
		<Synonym Name="FirstName" Term="FName"></Synonym>
		<Synonym Name="LastName" Term="LName"></Synonym>
	</Dictionary>

	<Dictionary Type="Subscriber" Name="ABCDE" Version="1.0">
		<Term Name="MyFName" TermType="Field" DataType="string" Length="20" />
		<Term Name="MyLName" TermType="Field" DataType="string" Length="20" />
	</Dictionary>
</Dictionaries>
```

9. MAPPING DEFINITION

The purpose of this data mapping file is to create a link between the data field terms defined in the data dictionary and the index position of that field in the MANIFEST.DXD file. A data map consists of an XML file named MAPPING.XML and will contain a set of dictionary terms along with the index value in the DXD record.

9.1 Format

This section describes the structure of the proposed data map file.

9.1.1 <DXDDataMappings> Element

This element is the root of the document, and may contain multiple <*DXDDataMap*> elements.

```
<DXDDataMappings>
       <DXDDataMap></DXDDataMap>
</DXDDataMappings>
```

9.1.1.1 Attributes

None

9.1.1.2 Nested Elements

This element will contain one or more <DXDDataMap> elements.

9.1.2 <DXDDataMap> Element

This element contains the definition of a collection of data mapping entries residing on a DXD file within the package. It may contain multiple <*MapEntry*> elements.

```
<DXDDataMap DXDFile="ManifestB.dxd">
       <MapEntry></MapEntry>
</DXDDataMap>
```

9.1.2.1 Attributes

This element has the following attributes

4) **DXDFile (Optional) :** If present, it will indicate the DXD File where the package data resides. Otherwise, it is assumed that the data is located in the MANIFEST.DXD file.

9.1.2.2 Nested Elements

This element will contain one or more <MapEntry> elements.

9.1.3 <MapEntry> Element

This element will contain the definition of a term and its index position in the dxd file.

```
<MapEntry Term="SSN" Index="3" />
```

9.1.3.1 Attributes

This element has the following attributes:

5) **Term (Required)** : This attribute contains the name of the term defined in the data dictionary.
6) **Index (Required) :** Index location of that term's value in the DXD file.

9.1.3.2 Nested Elements

None.

9.2 Sample file

The following is a sample of how a data mapping file would look like.

```
<?xml version="1.0"?>
<DXDDataMappings>
        <DXDDataMap>
                <MapEntry Term="FName" Index="1" />
                <MapEntry Term="LName" Index="2" />
        </DXDDataMap>
        <DXDDataMap  DXDFile="ManifestB.dxd">
                <MapEntry Term="SSN" Index="3" />
        </DXDDataMap>
</DXDDataMappings>
```

10. EXTENDED DOCUMENT DATA DEFINITION

Extended document data which is beyond that of what can be stored in the metadata file (MANIFEST.DXD) can be stored in an XML file named EXTDOCDATA.XML that contains a set of elements for any document file in the package. How application specific document data elements are used and defined is beyond the scope of this specification. Inclusion of this file in the package is optional.

10.1 Format

This section describes the structure of the Extended Document Data file.

10.1.1 <Documents> Element

This element is the root of the file, and may contain multiple <*Document*> elements. There does not have to be a <*Document*> element for every document in the package.

10.1.2 Attributes

None.

10.1.3 Nested Elements

This element will contain one or more <Document> elements.

10.1.4 <Document> Element

This element must contain one <Filename> element. Other elements may be contained in the <Document> element and are to be determined by specific application requirements.

```
<Document>
        <Filename></Filename>
</Document>
```

10.1.4.1 Attributes

None.

10.1.4.2 Nested Elements

This element must contain one <Filename> element. Other elements included are to be determined by specific application requirements.

10.1.5 <Filename> Element

This element specifies the filename of the document as specified in the MANIFEST.DXD file. The exact value of the "File Description" field (which is the filename) in the MANIFEST.DXD must be used. This filename is used as key to identify which document in the package the extended data belongs to.

```
<Filename></Filename >
```

10.1.6 Application Specific Elements

All other nested elements are application specific and their definition is out of the scope of this standard.

10.2 Sample file

The following is a sample of how a EXTDOCDATA.XMLfile would look like. This sample includes application specific elements for a document tracking application.

```
<?xml version="1.0" encoding="utf-8" ?>
<Documents>
     <Document>
          <Filename>230001.tif"</Filename>
          <VendorDocId>R3D7765</VendorDocId>
          <eX4TrackingId>CO1-0000012-1</eX4TrackingId>
       </Document>
     <Document>
          <Filename>230002.tif"</Filename>
          <VendorDocId>R3D7766</VendorDocId>
          <eX4TrackingId>CO1-0000012-2</eX4TrackingId>
     </Document>
     <Document>
          <Filename>230003.tif"</Filename>
          <VendorDocId>R3D7767</VendorDocId>
          <eX4TrackingId>CO1-0000012-3</eX4TrackingId>
    </Document>
</Documents>
```