eMortgage
PDF Guide for eDocs

Abstract

This MISMO® eMortgage PDF Guide, published by the Mortgage Industry Standards Maintenance Organization, Inc. ("MISMO"), a wholly owned subsidiary of the Mortgage Bankers Association, is a mortgage industry PDF reference tool, providing voluntary guidelines to standardize PDF based electronic documents.

Revision History

Date	Version	Description	Author
11/14/2005	1.0 Draft	Initial Working Draft	Ed Chase
12/16/2005	1.0 Draft	Changes per comments from eMortgage F2F	Ed Chase
01/16/2006	1.0 Draft	Alignment of outline with eDoc criteria	Ed Chase
01/27/2006	1.0 Draft	MERS eRegistry FAQ	Abbasi Najmi
02/24/2006	1.0 Draft	Third-party PDF libraries FAQ and other edits	Ed Chase
02/24/2006	1.0 Draft	Added Approved MISMO eMortgage Motion and conclusion in introduction	Igor Derensteyn
03/10/2006	1.0 Draft	Added PDF functionality matrix	Ed Chase

Table of content

1. Introduction

The MISMO SMART Document specification was developed to address specific requirements for electronic documents in the mortgage banking community. At the time SMART Doc was developed, no other electronic document format - including PDF (Portable Document Format) - was capable of supporting the complex XML technical requirements it was designed to address. Over time, since the development of SMART Doc, the state of electronic document technology has rapidly advanced. PDF now supports new capabilities that allow it to meet the XML document requirements of the mortgage banking community. Already, PDF has become a de facto standard for many mortgage related documents. The current versions of PDF can now deliver the additional functionality needed for standards-based and legally acceptable documents. This document provides guidance for the expanded use of PDF features to include direct support for those requirements specifically outlined in the SMART Doc specification.

Currently, MISMO is in a process of evaluating PDF as an additional voluntary standard format for electronically signed and data enabled documents in the mortgage industry. The following is a motion that MISMO eMortgage workgroup approved on November 15th, 2005:

> *MISMO eMortgage workgroup recommends the evaluation of PDF as an additional format for electronic documents, resulting in an implementation guide including requirements for:*
>
> - *describing,*
> - *visually representing,*
> - *embedded data within,*
> - *transparently linking data and visual representation relating to,*
> - *electronically signing,*
> - *tamper-evident sealing,*
> - *and auditing the document.*

The following is a link to MISMO minutes of the PDF proposal on November 15, 2005.
https://sharepoint.mismo.org/emortgage/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2femortgage%2fShared%20Documents%2feDoc%20Standards%2fPDF%20Proposal&View=%7b0C610FF3%2d739D%2d4414%2dACB9%2d4474F6782F1E%7d

Usability, simplicity, and low cost of electronic document formats will drive their adoption as long as the legal and operational requirements are satisfied. Today, PDF is already widely used format for electronic document exchange. The format is also widely used for electronic disclosures, appraisals, title commitments, flood determination reports, credit reports, and other electronic documents that do not require printing.

PDF format is evaluated because it may help to accelerate eMortgage adoption and supports an evolutionary approach from paper to imaged docs to esigned docs, and then to esigned docs with data while achieving incremental benefits.

2. **Key Concepts** - Summary of Key Standards and Technologies

Portable Document Format (PDF)
XML Data Package (XDP)
XML Forms Architecture (XFA)
XML Support and Format Transitions in PDF
PDF and Standards
The Publicly Available PDF Specification
PDF Digital Signatures

	Flat PDF		Data Enabled PDF	
	1. Flat PDF – Scanned Image	**2. Flat PDF –** Electronically Generated	**3. Data Enabled PDF -** with AcroForms	**4. Data Enabled PDF -** with XFA & XDP
Describing (Metadata)	*Through XMP or file attachments*			***XMP and native XML support***
Visually Representing	*Bitmap Image of Document*	*Vector & font-based image with selectable & extractable text*	*Form field annotation objects*	***XML template for layout and data fields***
Embedded Data	*Only with attachments*	*With attachments or through complex PDF-based processes*	*Through attachments, or embedded data in FDF or XFDF*	***Full XML Schema support***
Transparently Linking Data and Presentation	*Not available*		*Linking between Form Field Annotations and FDF/XFDF*	***Full linking of Schemas via XFA mapping***
Electronically Signing	***Fully supported with PDF Signatures***			
Tamper Evident Sealing	***Fully supported with Certification & PDF Signatures***			
Auditing	***Fully supported with Certification & PDF Signatures***			
Software & Libraries Available	***Adobe, 3rd Party Commercial, and Open Source***			
PDF/A	***Fully supported***		*Flattened only*	

Portable Document Format (PDF)

PDF is a file format for representing documents in a manner independent of the application software, hardware, and operating system used to create them and of the output device on which they are to be displayed or printed. A *PDF document* consists of a collection of *objects* that together describe the appearance of one or more *pages*, possibly accompanied by additional interactive elements and higher-level application data. A *PDF file* contains the objects making up a PDF document along with associated structural information, all represented as a single self-contained sequence of bytes.

A document's pages (and other visual elements) can contain any combination of text, graphics, and images. A page's appearance is described by a PDF content stream, which contains a sequence of graphics objects to be painted on the page. This appearance is fully specified; all layout and formatting decisions have already been made by the application generating the content stream. In addition to describing the static appearance of pages, a PDF document can contain interactive elements that are possible only in an electronic representation. PDF supports annotations of many kinds for such things as text notes, hypertext links, markup, file attachments, sounds, and movies. A document can define its own user interface; keyboard and mouse input can trigger actions that are specified by PDF objects. The document can contain interactive form fields to be filled in by the user, and can export the values of these fields to or import them from other applications.

Finally, a PDF document can contain higher-level information that is useful for interchange of content among applications. In addition to specifying appearance, a document's content can include identification and logical structure information that allows it to be searched, edited, or extracted for reuse elsewhere. PDF is particularly well suited for representing a document as it moves through successive stages of a prepress production workflow.

XDP and XFA
XDP and XFA are the XML specifications that describe the XML architecture used by the current versions of the PDF specification (1.5 & 1.6). Like PDF, the XDP and XFA specifications are publicly available. They extend PDF to support XML for a container, a template language, and a data model.

XML Data Package (XDP)

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.

XDP is not part of the MISMO PDF proposal. The summary is only for informational purposes. There are advantages of using XDP that may need to be evaluated going forward.

XML Forms Architecture (XFA)

While XDP is an XML container for the XML expression of PDF, the XML Forms Architecture (XFA) provides a template-based grammar and a set of processing rules for the description of interactive forms. XFA is the layout, event, and data template for XML-based PDF. It provides layout capabilities for complex and precise page-based forms, an event model that translates into PDF, and the data binding, validation, and masking structure used to interact with XML schema-based vocabularies. An XFA template describes how an interactive form should appear and behave. It can play a role in several situations: interacting with a user, printing forms, and processing machine-generated data.

XML Support and Format Transitions in PDF

Through XFA and XDP, PDF can interact with XML in two fundamental ways:

1) PDF can serve as a host for XML vocabularies (like the MISMO XML data standards). PDF can be a container for XML and supports mapped view and data, interactive validation, and import/export of XML instances. Mapped view and data is addressed through the logical mapping of an XFA-based view template to an XML instance or schema. Interactive validation allows PDF viewing applications to enforce XML and higher level data requirements during interactive form-filling. XML import and export means that the PDF can both import and export valid instances of any pre-configured XML instance or schema.

2) PDF can be represented as XML. An XML-based PDF can also be represented in an XML format (XDP). XDP is a verbose, plain-text XML version of the PDF containing template, binding, instance data (all represented in XFA), and potentially a binary-encoded PDF with digital signatures. In XDP format the file is encoded in plain-text XML that can be processed by any application that supports XML. Instance data is directly available under the Data node within the file. The XDP to PDF (and reverse) transition preserves structures like template, data, and mapping.

The transition, import/export, and other capabilities above can be performed by many PDF-aware applications. The ability to perform these functions is not restricted to Adobe products, and is replicable by third-party commercial or open source software as

indicated in the PDF specification.

The following is a link to one of third-party libraries to demonstrate how to get XML data from PDF (no XDP) using the third-party APIs as an example:
http://big.faceless.org/products/pdf/docs/api/org/faceless/pdf2/XFA.html

PDF and Standards

PDF support for XML means that PDF documents can now support standards-based XML from most commercial and public sector XML standards efforts. XML Standards Development Organizations that have formally acknowledged or directly integrated XML-based PDF into their standards include: ACORD - Insurance Forms; UN/CEFACT - United Nations International Trade Documentation; and RosettaNet - Manufacturing and other B2B applications. PDF is a logical format of choice as platform for XML standards since it can support nearly any XML-based standard, and yet provide easy access to documents through a large installed base PDF software.

The Publicly Available PDF Specification

There is considerable confusion and much opinion about the nature of the PDF specification. The PDF specification is published by Adobe, and has been for roughly ten years. This means that any developer - individual, public sector, or business - can, without restriction, obtain and implement PDF. Adobe does not restrict or control 3rd party development with PDF. As a result, there are thousands of developers now using the specifications and working with PDF. Adobe has updated the PDF reference with every major release of its Adobe Acrobat product family. The specification is updated to include support for new functionality - as new requirements are addressed through PDF, they become a part of the public specification.

The availability of PDF has also resulted in the actual development of true "open" standards around it. There are now two ratified ISO (International Standards Organization) standards. PDF/Archiving (PDF/A) for the long-term preservation of electronic documents was driven by groups including the US Courts and US National Archives and Records Administration, and was ratified in 2005. PDF/X (REFERENCE) for reliable pre-press exchange was ratified in (DATE). Currently under development are also PDF/E for engineering documentation and PDF/UA for reliable access by assistive technology.

PDF Digital Signatures

A digital signature can be used to authenticate the identity of a user and the document's contents. It stores information about the signer and the state of the document when it was signed. The signature may be purely mathematical, such as a public/private-key encrypted document digest, or it may be a biometric form of identification, such as a

handwritten signature, fingerprint, or retinal scan. The specific form of authentication used depends on the implementation requirements. The discussion of digital signatures in this document specifically refers to PKI (Public Key Infrastructure) signatures in PDF. PKI signatures are the default PDF signature mechanism most PDF signature applications. They provide significant technical and legal advantages beyond image-only and "clickwrap" type signatures. PKI signatures can be used to definitively establish identity (individual or organizational), to detect and/or prevent modifications to documents (tamper-evident sealing), and to establish secure audit trails. PDF signature functionality is discussed in detail in several parts of section 3 below - e) eSignatures, f) tamper-evident sealing, g) audit trails, and h) document integrity.

3. Guidelines

a. Metadata
b. View
c. Data
d. Mapping
e. eSignatures
f. Tamper-Evident Seal
g. Audit Trails
h. Document Integrity

a. Metadata

Metadata is data that describes the characteristics or properties of a document. It can be distinguished from the main contents of a document. For example, for a word processing document, the contents include the actual text data and formatting information, while the metadata might include such properties as author, modification date, or copyright status. There can be gray areas where the same information could be treated as content or metadata, depending on the workflow. In general, metadata should have value on its own without regard for the content. For example, a list of all fonts used in a document could be useful metadata, while information about the specific font used for a specific paragraph on a page would be logically treated as content.

In the case of MISMO, the concept of metadata has applications that might include:

File Format Metadata. This is the information that is the most specific to a particular file format (PDF, HTML, SMART Doc). This metadata might include document format type & version, creation, modification, and signature dates, or other items that apply to any document of that format. The information here describes a document in the broadest sense, and should be available to generic viewing applications. In the case of PDF, this information would apply to all PDFs.

Process/Domain Metadata – In the case of eMortgages, these elements are those that pertain to all mortgage documents. They are mortgage specific, but are used across all documents. Elements like document type, DTD version, and document state would be considered process/domain data. In the case of mortgages, these elements apply to all mortgage documents.

Document Specific Metadata – This is the most specific to a document type, and includes all elements that are neither generic to a file format nor process-specific. In the case of mortgages, it includes elements from the individual MISMO process group data standards, and reflects the assembly of standard data elements from the MISMO data dictionary into document specific schemas. This dataset is specific to a single mortgage

document type, though may be a subset of a larger dataset.

How these different concepts are serialized in PDF depends on the goals of the implementation. PDF can support metadata in two different ways - with XML Data and with XMP:

Metadata with XML Data
Most Process/Domain, and Document Specific Metadata (the second and third types from above) should reside within the underlying XML dataset in the PDF (see section 3-d for information on data mapping concepts). This preserves interoperability with other XML-based documents and systems, and is easiest to implement in PDF. In this case, the metadata would be addressed in the underlying MISMO XML DTD, and mapped (or not) as appropriate.

Metadata with XMP
XMP is a universal RDF/XML-based framework for File Format Metadata (the first type from above) in PDF. It can be used for basic file metadata like authoring applications, users, dates, and system identifiers. However - XMP is an extensible framework - it does not restrict the types of metadata it can contain. This means that a single metadata packet can contain information from any number of metadata vocabularies - image, document, workflow, content management, ownership data, or any custom defined set. Organizations can use XMP as a framework for their own metadata elements. The addition of new elements to an XMP packet will not break compatibility with other systems. XMP also allows the cross-referencing of information. Metadata elements that might be shared by identification, workflow, and content management metadata can all reference the same data. This cross-referencing reduces duplication and improves portability. Finally, XMP is serialized as plain-text within a PDF file. This means that applications do not need to be PDF-aware to access and process the metadata within the file.

The decision to place individual metadata elements in XML or XMP depends on the implementation requirements, though for eMortgages in general, most relevant metadata is best addressed through the underlying XML dataset, with only file format-specific metadata addressed through XMP. This preserves interoperability with other XML-based systems and is simplest to implement.

b. View

PDF is a file format for representing documents in a manner independent of the application software, hardware, and operating system used to create them and of the output device on which they are to be displayed or printed. PDF documents provide a high level of precision and consistency among different platforms, and between printed and electronic versions. The XML Forms Architecture (XFA) provides a template-based

grammar and a set of processing rules for the description of interactive forms. XFA is the layout, event, and data template for XML-based PDF. It provides layout capabilities for complex and precise page-based forms, an event model that translates into PDF, and the data binding, validation, and masking structure used to interact with XML schema-based vocabularies. An XFA template describes how an interactive form should appear and behave. It can play a role in several situations: interacting with a user, printing forms, and processing machine-generated data.

Interactive Views – Validation, Masking, and Scripting
- Validation provides a mechanism for establishing that form values are of a certain type, or within a certain range.
- Masking is the ability to modify the format of a field between what the user sees, and what is contained in the dataset.
- Scripting provides the capability for interactive calculations and more advanced features like dynamic forms or complex logic.

Validation
Validation provides a mechanism for establishing that form values are of a certain type, or within a certain range. It provides the capability to compare user input to those values that meet the requirements of a particular data element. Validation information in XML-based PDF can be derived directly from the underlying XML dataset and/or manually added by the author of a form. Basic validation in interactive forms is not dependant on scripting, though scripting can be used to employ advanced validation techniques, including accessing web services or other outside resources.

Masking
Masking is the ability to alter the format of a field between what the user sees, and what is contained in the dataset. SMART Doc has a very similar capability. A simple example of masking can be demonstrated with calendar dates. In human-readable PDF, it might be desirable to display a date as: "January 1, 2005". However, a more easily machine-readable XML date might be serialized as: "01-01-2005". Masking allows information to be formatted separately between presentation and data views. As in SMART Doc, masking in PDF does not have the capability to alter actual data. It simply provides alternative viewing formats. Masking in XML-based PDF is built directly into the presentation specification. Scripting is not required for masking.

Scripting
One of the most frequently asked questions around the use of PDF for high value documents has been the question of scripting. Scripting has value in many uses in PDF - it enables the development of more intuitive, interactive, and powerful PDF documents. Questions about the use of scripting for high value documents tend to center around the risk of content modification between the data and presentation layer, or modification of a document after it has been digitally signed or approved.

Though authenticity mechanisms exist in PDF for protecting document integrity - even when employing scripts, it may the best practice to severely restrict or limit the use of scripts within certain high value documents. The ISO standard PDF/A for long term document preservation actually prohibits any scripting in compliant PDF files for exactly the reasons above. In these cases, scripts may be more securely employed outside of the document - at the folder or application level. There are also features in PDF that can make the use of scripts less apt to generate apprehension in these scenarios. Two features in particular - Author Signatures (or Certification) and explicit security restrictions in PDF scripting on certain methods go a long way to address the risk of script abuse. Organizations need to evaluate the potential benefits of scripts against the potential liability on a document or process basis. In most cases, the higher value the document, the more cautious implementers need to be with the use of scripting.

Static or Dynamic Documents

The more recent versions of the PDF specification are capable of supporting more flexible document structures. This flexibility allows the development of documents that can expand or otherwise modify themselves based on business logic or the expansion or addition of XML data nodes. The simplest example of this would be a document with multiple repeating line-items - like a list of comparable properties in an appraisal. A dynamic PDF could expand to accommodate any number of properties, based on repeating XML nodes. This means that the document author doesn't need to decide exactly how many entries might be required, and users of the document don't need to manually add continuation pages. As more items are needed, the PDF is automatically re-rendered from the underlying template to accommodate additional data. This process can be driven automatically by the underlying XML dataset, or can be done through scripting to allow user input or business logic to trigger the modifications.

While dynamic documents have definite appeal for certain types of content, other, more high-fidelity or high-value documents may need to have a more consistent and static structure - even the smallest modification to legally binding or negotiable documents may not be desirable for high-value documents. In these cases, a static PDF structure allows more presentation consistency, and is less likely to create concerns around the final or legal state of a document.

The PDF specification addresses the need for different levels of document presentation flexibility. To provide a capacity to distinguish between these document types, implementers can selectively create either "Dynamic" or "Static" PDF for XML-based forms. These parameters are configurable during the PDF generation process, or through the authoring application. A Dynamic PDF permits re-rendering and expansion of a document for interactivity. Dynamic PDF can even be used with document Certification signatures to authenticate the scripts and logic behind the dynamic portions (see Author Signatures and Certification in the next section). Even so, there are still requirements for high-value static documents. For this, Static PDF does not allow the re-rendering of it's content. These two designations allow for the differentiation between traditional static

PDF for high-value documents, and dynamic documents for more interactive and flexible applications. Which type of PDF is generated is dependant on the rendering application. The merits of each type should be evaluated by implementers.

c. Data

PDF supports the MISMO Data standards. PDF documents can be a platform for industry data standards (like MISMO XML data standards). At design-time, by way of a "binding" process, a PDF can be mapped to a particular XML Schema or instance document. The binding process creates links between data fields on the PDF and elements and attributes within an XML document. When configured in this way, XML instance data can be imported to populate the document automatically, or data can be entered manually by users. When data is exported to an XML file, the XML file will conform to the XML schema or instance bound to the PDF.

Since the PDF can be represented in multiple formats (PDF & XDP), it's important to understand where the XML instance is physically located.

In PDF - The XML instance is contained within the PDF. In PDF state, the document will need to be processed (in accordance with the PDF reference) to obtain the XML instance document. XML nodes can be referenced through the PDF object model, but the PDF can not be viewed as plain text to view the XML.

In XDP - When a populated PDF is represented as XDP, the XML dataset is directly accessible in plain-text XML. No PDF processing is required for access to the XML data in an XDP. When a populated PDF is represented as XDP, and a binary-encoded copy of the PDF is included (as in a digitally signed document), the XML data will exist in both the XML plain-text portion of the XDP as well as inside the binary-encoded PDF.

d. Mapping

At it's core, the MISMO concept of an eDoc is a platform for human-readable view (documents) and machine-readable data (XML data standards). One of the main goals of MISMO eDocs is to address the relationship between those concepts - view and data. To this effect, the SMART Doc specification includes a powerful mechanism for "mapping" or relating data elements to presentation elements. This mapping mechanism ensures that users and machines are viewing and acting on the same data, and helps ensure the legal validity of high-value electronic documents.

XML Binding
The PDF specification now includes support for a view-to-data binding mechanism (see the XDP and XFA supplements at the end of this document for more information). PDFs can now support mapped relationships between human readable documents, and machine readable XML. Prior to the PDF 1.5 specification, XML data support in PDF was

somewhat limited; XML could only be derived in a single format, known as "XFDF" (XML Form Data Format). Conversion of XFDF to another XML vocabulary required the use of stylesheets, which meant that the final data set was not directly mapped to the original presentation. Version 1.5 of the PDF specification added the ability to "bind" PDF form elements directly to XML structures, elements, and attributes of any DTD or XML Schema. This allows PDF to now directly support many industry data standards, including the MISMO XML standards.

To generate these XML-based PDFs, the view section of PDF forms and documents are "bound" to an XML schema. Binding involves the mapping of the data element to the presentation element. Binding statements exist in the presentation section - effectively "pointing" to an XML node with an XPath like statement. Binding to arbitrary XML is now intrinsic to PDF; it doesn't depend on post processing, transformations, or scripting.

An actual data binding from a PDF form to an XML data set:

"$record.ApplicantInfo.OrganizationInfo.Address.ZipCode"

Bindings can be single or multi-directional, by controlling the read-only status of the bound field. One directional binding would mean that the form field was not interactive; it simply displays the bound XML in the linked field. One directional binding scenarios might involve pre-population of the XML by an application, then displaying the completed document to a user. Two directional binding is functionally identical within the structure of the PDF, but fields are no longer read-only. Data can be entered into interactive fields by both users and systems. Changes made by a user are reflected directly in the XML dataset. Both types of binding, one-directional and two-directional can exist in the same document. Many mortgage documents have some sections that are not intended to be modified by end-users (like interest rate values) and other sections that may require additional information from the parties at closing (Notary data).

PDF support for view-to-data binding means that, like the SMART Doc, presentation and data are contained in a single file. Unlike SMART Doc, there exists only one definitive dataset in PDF, which is used for both static and interactive forms. When the PDF is rendered, XML data is obtained directly from the underlying dataset, and rendered with the view. Since there is only one dataset, validation of the document by comparing the information in the view and data sections is not required - the same data is used by both.

e. eSignatures

ESign and UETA provide the legal foundation for electronic transactions, but stop short of mandating any particular technology or type of technology. They provide a baseline for the requirements on which implementers can build compliant platforms. This allows technology to advance without making the laws obsolete. The use of PKI-based signatures in PDF is one of the major aspects of the format that directly supports the

policy requirements designed to comply with UETA and eSign. While the use of a technology or document format alone will not actually provide eSign/UETA compliance by itself, it can support those policies by providing significant assurances as to the integrity of documents, the identity and intent of signing parties, and the non-repudiation requirements for legally binding electronic workflows. PDF provides all of this capability, and when properly implemented along with appropriate accompanying policies and procedures can fully support eSign and UETA compliant transactions.

PDF as a Legal Document
PDF is widely accepted as a legal document. PDF is mandated or specified by organizations like the US Courts (Case Filings), the US Patent and Trademark Office (Patent Applications), and the Food and Drug Administration (New Drug Applications). The presentation fidelity and document integrity characteristics of PDF make it especially applicable to legal documents. PDF also provides mechanisms for advanced document authenticity, in the form of PKI-based digital signatures. Digital signatures in PDF can be used by authors to sign original documents before initiating a workflow, by recipients to acknowledge, approve, or agree to the terms of a document, and to audit the lifecycle of a document - each signature in a PDF creates a "snapshot" - a copy of the version that was signed. Since modifications to a PDF are appended to the end of the file structure, previous signed versions of a PDF are always accessible - even after multiple signatures. This information can be used to visually identify alterations between or after signatures, or to "roll back" a document to a previously signed state.

f. Tamper-Evident Seal

PDF includes robust support for certificate-based digital signatures. With respect to modification detection, digital signatures in PDF are synonymous with the SMART Doc concepts of "Tamper Evident Sealing". They provide proof that a document has not been modified since a signature was applied, and they can uniquely bind a signed document to a particular digital identity. Each signature in a PDF provides a tamper evident seal on the document. Digital signatures in PDF are standards-based (x.509, PKCS), and can provide robust authenticity for documents. They also support secure time-stamping, full path validation and revocation checking for certificates. PDF is able to natively support the use of SISAC credentials, fully supporting 3rd party and Certificate-Authority based PKI.

Certification
Certification (or Author Signatures) allows the author of a document to authenticate it before distribution by signing the layout, data bindings, logic, and scripts inside a document. Certification signatures give recipients assurance that a document was approved by the author and has arrived unaltered, and they ensure that when the sender receives a completed form or signed document at the completion of a workflow, that none of their scripts, bindings, or logic have been altered by recipients. Certification prevents recipients or other parties from inserting malicious scripts that might alter calculations or values, modifying the text of a contract, or attempting to hide text in small

or page-colored fonts. Certification parameters can be customized by the author to selectively allow form-filling, scripting, or interactivity with the document. Certification also provides a mechanism to notify recipients of any scripts or interactivity contained inside a document. Recipients are automatically alerted if a certified document allows form-filling actions, or if it contains any scripts. Finally, certification allows authors to bind scripts to the certification status of a document. Certain security-sensitive scripts can only be run in certified documents where the recipient has explicitly trusted the author.

Author Certification is highly recommended for any documents that are expected to be signed by recipients, documents that may become legally binding, or documents of record. It protects both authors and recipients, ensuring that content can not be maliciously or accidentally altered over the lifecycle of the document. In the context of SMART Doc, certification provides the functionality that to the validation of SMART Doc view and data sections does, albeit in a different manner. Instead of requiring post-process validation of these sections manually, PDF provides the ability to for authors to authenticate the document from the start, protect the content throughout the entire process, and re-authenticate the certification upon completion to establish validity.

g. Audit Trail

Secure audit trails have two fundamental requirements:

- Audit Data - Recorded information about the date/time and nature of an auditable event.
- Audit Authenticity - Assurance of the integrity of audit information of a document.

Audit Data
Since audit data is generally domain-specific - i.e. it concerns events that have particular meaning in a certain context - "signed", "populated", "tamper-sealed" - audit data is generally a member of a particular XML vocabulary. Support for industry-specific audit data in PDF is covered through PDF support for XML data standards (see section c. Data). A PDF can contain any audit data as part of its underlying XML instance.

Audit Authenticity
Audit authenticity can take several forms - it can be provided by a system-specific mechanism, or through the built in capability of PDF to produce a secure audit trail using digital signatures. PDF files support the use of multiple and sequential digital signatures to ensure document authenticity. Each signature on a PDF document produces an auditable event - and can include information like secure timestamps and certificate revocation responses - to establish exact event chronologies. The structure of PDF is based on an incremental update capability which lends itself well to multiple signature workflows. As a PDF is modified, the modifications to the document are added to the end of the file. When the document is rendered, the modifications are integrated into the current rendering - preserving the previous document states. This incremental updating is transparent to the person viewing the document, but when combined with PDF digital

signatures, allows for the detection and audit of modifications to the file. PDF applications that support digital signatures may allow users to view the state of a document at the time each signature was applied. Advanced PDF tools may use the audit information to visually identify and report on document modifications.

h. Document Integrity

Document integrity in PDF is a function of many of the previous sections - particularly those that deal with digital signatures, with some consideration toward understanding dynamic PDF capabilities and scripting. PDF digital signatures provide a strong, PKI-based integrity mechanism for documents. Modifications to a digitally signed PDF are intrinsically detectable, and auditable. A document author may choose to limit what levels of interactivity or form filling capability are acceptable for a particular workflow with Certification signatures. Certification signatures in particular are critical to document integrity. Certification signatures authenticate a document before users interact with it. They can eliminate the need for additional data validation by ensuring document integrity throughout a workflow. Certification signatures also enable better multi-signature workflows since they reduce the need for individual signature validation.

See also: Sections 3-f: "Certification"; and 3-b: "Scripting" and "Static or Dynamic Documents"

4. FAQs

4.1 How open is the PDF specification? *Are the specifications freely available for use by everyone, without any restrictions (e.g., through a license agreement, patents that cover the technology, etc.)? For example, are Adobe's competitors free to use the specifications to create competing products?*

The exact IP statement of the PDF specification can be found in section 1.5 of the PDF Reference (also see 4.2 below). The PDF, XDP, and XFA Specifications are all freely available for use by anyone. Adobe's competitors can (and do) to use them to create competing products. It's important to differentiate here the difference between PDF and Adobe Acrobat/Reader. PDF is a specification for a file format, Acrobat and Reader are products - the IP around which is separate from that of PDF. There are SDKs and APIs for enhancing and extending Acrobat and Reader, but there are significant differences in IP when dealing with Adobe products as opposed to PDF.

PDF is not the only specification available from Adobe. We know TIFF as a "de facto" standard in the imaging world. TIFF specification has been administrated by Adobe, since Aldus merged with Adobe in 1994.

Also, PostScript is another "de facto" standard in the printing world, and it is a 20+ years old specification from Adobe. Leveraging PostScript, Adobe introduced PDF format in 1992, and it has been esignature enabled since 1999.

4.2 How we can confirm that the specifications will continue to be freely available and provide us with back-up documentation?

Specific, written permission has been granted to standards development organizations (AIIM) to freely publish the PDF specification. The goal of this has been to alleviate any concerns around the continued status of the PDF specification in the event of fundamental changes to, or the absence of Adobe Systems, Inc.. The granting of this permission has been instrumental in the development of formal PDF-based ISO standards, and formalizes the long-term availability of the PDF specification.

4.3 Does Intelligent PDF 1.6 specifications mention embedded technology (e.g., digital signatures) that might be subject to licensing fees? *Elaborate on these instances and provide copies of any licensing documentation?*

The PDF 1.6 and Intelligent Document (XFA, XDP) specifications include specifications for digital signatures. To implement these features from the specification does not require any license fees. vThere are a number of third parties that provide digital signature

functionality in PDF. Third parties are not limited to create or manipulate digital signatures within their own products.

4.4 Do you know of any current instances where a legally binding contract has been executed using the Intelligent PDF format? Please provide details.

UN/eDocs – negotiable waybills; ACORD – insurance contracts; USPTO – patent applications; FDA – new drug applications; HSPD-12 – credentialing; SAFE – pilots for pharmaceutical samples.

4.5 Technically, how does the rendering of the document occur?

4.5.1 The data and the view are housed in one file, correct?

Correct. Data and view are contained in the same PDF. The PDF can be represented as both a traditional PDF document, or also as an XDP file. An XDP is a verbose XML document containing: XML versions of the template/view; the XML Schema of the document; binding information linking view and data; scripts, masking, and localization information; other arbitrary XML content as required.

4.5.2 There is only one database, correct?

Correct. There is a single XML data set that may be enveloped within the PDF, or exposed as XML in XDP.

4.5.3 Is it a scripting process that renders the image? Or browser initiated?

Rendering is not script driven. There are two types of rendering involved with Intelligent Document PDFs.

The first is the rendering of the template – the XDP rendered to PDF. This does not typically occur at the client (Acrobat or Reader) but usually occurs at design time – the PDF is rendered from the XFP by a desktop or server application. Server-based tools are typically used to render PDFs in situations where a previous part of the process might result in an individualized document based on earlier system or user input. In either case, this rendering process may be reversed – A PDF generated from an XFA XDP can be reconstituted as an XML XDP file. If data has been added to the PDF since it's rendering, it will be reflected within the data node of the XDP. The rendering process is specified in the XFA and XDP references (publicly available and subject to the same IPR as the PDF reference).

The other type of rendering that occurs is when the client application (such as Acrobat or Reader) displays the PDF for viewing. This rendering process is detailed in the PDF specification and can be compared to the rendering of any electronic document or image for viewing in an application.

4.5.4 How does it compare to the XSLT technology?

XSL-T is used to transform XML from one XML format to another. The possibilities with XSL-T are quite open-ended, but it's typically used to format data-oriented XML of a particular format into XHTML for viewing. Most XSL-T transformations can be regarded as one-way, in that they result in documents that can not be reliably returned to their original state. This one-way transformation means that additional verification and/or multiple files may be necessary to authenticate view against data. XDP-to-PDF and PDF-to-XDP transformation is unlike typical XSL-T transformations in that the transformation between PDF and XDP formats can be bi-directional. A PDF can begin it's lifecycle as an XDP, be rendered as a PDF for user interaction, and then be returned to an XDP for XML processing.

4.6 What "tamper evident" technologies does PDF support and compare/contrast these technologies with MISMO SMART Doc standards for Tamper Evident Digital Signatures?

PDF has robust features to support the tamper-evident sealing of documents. In many ways, these features are functionally similar to those in SMART Doc, but they add additional capabilities in the areas of auditing and long-term validation. PDF tamper sealing is based on PKI digital signatures. Digital signatures in PDF can be used by users to approve or agree to the content of a document, or by organizations to certify the origin or validity of a document, or by systems or users to tamper seal documents. Each of these functionalities are based on the underlying PKI signature framework in PDF. In fact each time any digital signature is applied to a PDF, it creates an auditable event and effectively tamper-seals the document. In a process that contains multiple signatures – whether specifically for tamper sealing or not – creates a tamper seal. In multiple signature processes each subsequent signature "wraps" the document, data, annotations, attachments and all previous signatures within it.

Digital signatures can be applied with a number of PDF applications. These applications can often use the signature information in PDF to inform users about the validity of the document. They can validate that a document has not been changed since a signature was applied, and can "roll-back" the state of a document to view the signed state – before modifications were made. Advanced PDF tools may even create visual reports that highlight changes between signed version. PKI digital signatures in PDF are largely

based on existing standards – x.509 certificates and PKCS standards for cryptography and signature information. Signature information in a PDF is contained in a PKCS #7 standard block. There are a number of PDF tools that offer the ability to access this information.

PDF digital signatures also support advanced long-term validation. Long term PKI signature validity can be problematic with issues like credential expiration, or time/date ambiguities. Long term signatures in PDF support the embedding of secure time-stamp information and revocation responses within a PDF signature. This can provide definitive information about the time and date a signing even occurred, and an authenticated copy of the revocation response obtained from a CRL or OCSP server at sign-time. Since this information is embedded in the PDF, and authenticated by the signature, it can authenticate the document even after the expiration of credentials or without access to the credential's certificate authority.

4.7 Compare the Intelligent PDF technology to SMART Documents in terms of the ability to carry/transport other documents types.

PDF can be a container for other document types. PDF can support attachments in PDF or in any native format. PDF attachments can be page-level or document-level. Page level attachments are referenced by an annotation – an icon added in a layer above the PDF. Clicking on the annotation in a PDF reader will typically launch the application that created it and open the document (if available). Attachments can be added or extracted on the desktop, or by enterprise server applications.

PDF and PDF digital signatures can be used to secure and authenticate attachments. When digital signatures are applied to a PDF that contains attachments, the attachments are included in the processing of the signature. Modification to attachments in a signed PDF will be indicated and can be used to invalidate the signatures of the PDF. Similarly, encrypted or rights-managed PDFs also protect the confidentiality of their attachments, preventing access to them unless the PDF is opened by an authorized user.

4.7.1 Are XDP and XFA openly available specifications as well?

The XFA and XDP specifications are publicly available. They are subject to the same IP standards as the PDF reference.

4.8 How fully does Intelligent PDF technology support MISMO XML standards?

Intelligent PDF Documents fully support the MISMO data standards. XFA supports XML Schema for XML data models. The MISMO XML DTDs can be losslessly

converted to XML Schema with off-the-shelf XML tools.

4.8.1 Do the Adobe PDF specifications use standard MISMO terminology? If not is it available via ISO?

The PDF specifications employ ISO terminology.

4.9 How similar does the Intelligent PDF document handle the audit trail and header capabilities as outlined and managed by MISMO SMART Document standards?

Audit trails can generally be of two methodologies in PDF. In the most secure method, audit trails are directly linked to digital signatures. As outlined above, PDF digital signatures include built-in audit trail functionality. Each signature event is a definitive and auditable event. PDF can also support more generic audit metadata. PDF and XFA supports un-bound XML data that can be used for workflows and audit trails, but is not displayed in the document. XDP is also extensible – it supports arbitrary XML outside of XFA or PDF content. Arbitrary XML from an XDP can be contained in a rendered PDF and is reconstituted when the PDF is re-transformed to XDP. In the case of MISMO standard audit trails, they can be added and manipulated as arbitrary XML in a PDF/XDP.

4.9.1 Does the header section allow for transporting metadata? Does it allow for transporting of metadata that might not be displayed in the view?

Arbitrary XML like the MISMO header info can be retained in the PDF and XDP in sections outside the data section. Parts of the data section can also serve as hidden metadata – they just wouldn't be bound to view elements. PDF also supports it's own built-in metadata framework – XMP. XMP is a generic RDF-XML based metadata framework designed to support metadata standards like Dublin Core, EXIF, SCORM, etc. XMP doesn't specify metadata values beyond a base set (DCMI) and can be readily extended to support any required metadata standard.

4.10 How do the multiple signature capabilities with the Intelligent PDF versus those available via SMART Documents?

See the answer to question 7 for more signature info. Multiple signature workflows in PDF are easily implemented. Each signature in a PDF creates an auditable event, and "wraps" the document and all previous signatures. As a result, signed PDF documents contain built-in signature audit trails and metadata, and can be "rolled back" to the state when a signature was applied. SMART Doc/XML dSig implementations are capable of

similar implementations, though they would require very specific detailing of a particular document format. The generic structure of a PDF and built in support for this type of workflow makes it simpler to use. To perform the same comprehensive auditing on a SMART Doc would require implicit DOM information for a particular SMART Doc structure on a one-by-one basis. PDF signatures can take advantage of the PDF/XFA DOM to extract specific modification info about any PDF.

4.11 Technically, how are scripting controls identified and managed? Who manages the controls? *Is this something that implementers and others would need to build controls around if the PDF Intelligent document was accepted as an additional standard?*

Scripts are generally attached to events of objects in the XFA DOM. They can also exist as independent script objects. All scripts in an XFA-based PDF are visible in the XDP document. Additional scripts can not be added after rendering to PDF. All scripts are tagged with the header “JavaScript” or “FormCalc” depending on the scripting language. This means that an XDP can be audited for scripts by processing the XML in an XDP, making scripts are immediately identifiable.

Script use is also flagged during document certification. Certification signatures give document authors the ability to sign documents (including scripts contained within) before distribution. This prevents accidental or malicious modification of documents or their scripts during workflows. Certification protects both document authors and users. During the certification process, document scripts are automatically flagged, and an alert is automatically generated, notifying users that the document contains scripts, and the potential consequences thereof.

Scripts are not generally critical to the basic functionality of a PDF. They can provide enhancements to user interaction, dynamic document content, or complex validation, but they are not required for binding/mapping, rendering, masking, basic validation, signatures, or the import/export of XML. Scripts can be used safely, provided sufficient precautions are taken. Users are not easily able to add or modify scripts in the completed PDF. Adding certification makes it nearly impossible for anyone other than the document author to modify or add scripts to a PDF document. Still, for high-value or negotiable documents, the use of scripts might not be recommended, since they can give the perception that the state of a document might be altered after signing. Notes and Security Instruments might be considered of sufficient risk and value that scripting may need to be restricted or disallowed.

4.12 Can PDF support attribute value field bindings? *For instance, can a field in a PDF be bound to a single and exact repeating element based on the value of one of the element's attributes?*

There is a way to align attribute values with the binding of fields in XFA. It allows for the substitution of an attribute value for the element name in the binding. It's referred to in XFA as a type of "Extended Mapping". Full details are found in the XFA Data Handling sub-specification section 413 (http://partners.adobe.com/public/developer/en/xml/data_handling_2.0.pdf)

Basically, it allows the placement of a substitution statement in the config section of an XDP that looks something like this:

```
<transform ref="RESPA_HUD_DETAIL">
  <nameAttr>_SpecifiedHUDLineNumber</nameAttr>
</transform>
```

Then the individual field binding references would be set to something like this:

$record._CLOSING_DOCUMENTS.1001._LineItemAmount

This will effectively sort and force incoming data from an XML stream into the appropriate lines, based on the specified attribute (_SpecifiedHUDLineNumber in this case). No XSL-T or preprocessing is required with this method. It's very lightweight, and is fully supported in XFA.

In a SMART Doc binding, an equivalent XPath expression might look like this:

//RESPA_HUD_DETAIL[@_SpecifiedHUDLineNumber='1001']/@_LineItemAmount

The two methods are functionally equivalent; the XFA one just front-loads the binding configuration, where the XPath one spells the whole path out for each individual field.

4.13 Where do signatures reside in an XDP at the end of a workflow? *Are they reflected in the XML instance? What about the status of the template? Does it reflect the final status of the form?*

In addition to template, schema, and instance data, a completed XDP can contain a complete copy of the PDF encoded as base-64 within one of it's nodes. When a PDF is converted to an XDP after signing, this is where the signature information resides. The template in the XDP section is still the base form template and has not been modified.

The reason for this is that the encoded PDF combines the data and view, and is effectively "what the signer saw". It provides a higher level of document validation than the re-rendering and re-merging of the XFA template and dataset. Also, it provides the PDF capability of establishing a continuous audit trail and tamper-seal with each

signature.

This does result in a co-location of data within the XDP between the plain-text XML instance in the XFA data node and the XML instance within the encoded PDF. The signed version in the PDF can be compared to the instance in the XDP for additional validation, though the validation of the PDF signatures alone is enough to validate its own copy of the instance.

4.14 What additional guidance is provided by ISO PDF/A for long term document preservation?

The PDF/A specification is available from ISO for a fee:

http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=38920&ICS1=37&ICS2=100&ICS3=99

It provides very specific guidance around long term preservation of documents. Much of the guidance is very targeted at final-form documents that are no longer interactive. It is based on an older version of the PDF specification (v1.4) and does not deal with many of the newer features of PDF (including XFA). Some general guidance that it provides is:

- Fonts: Font embedding is encouraged as required document fonts may be unavailable, producing inconsistent rendering. Font embedding (full or subset) eliminates this dependency.
- Encryption: Encryption of final-form documents is discouraged because of the potential for permanent loss of information in the event of lost encryption keys.
- Multimedia: Since multimedia content relies on external software to function and is not part of the core PDF reference.
- Color: Color must be defined within a particular color space.
- Lossy Compression: Lossy image compression like JPEG is discouraged.
- Transparency: Transparent objects should be flattened to ensure consistent rendering.
- Annotations and Scripts: Multimedia and other file attachments are prohibited. Scripts are severely restricted.
- Metadata: XMP metadata is encouraged.
- Digital Signatures: Digital signature fields must be non-interactive.

4.14 Does the MERS® eRegistry support eDoc in the PDF format?

The short answer is, yes it does. The remaining section of this document will explain how MERS® eRegistry can be used to support PDFs.

There are two ways to register eNotes with MERS. One is with the presentation of a SMART Document and one is without the presentation of the SMART Document. The second method is the one that can be used to register eNotes in the PDF format with MERS® eRegistry. Please check with your investor.

Data Section:
When a MIN Number is presented for registration on the MERS® eRegistry, the following data points (listed as an xml section) are needed, in the form presented to register the MIN. The section presented here is a part of the registration request.

```
<_DATA_INFORMATION>
        <DATA_VERSION _Name="" _Number=""/>
</_DATA_INFORMATION>
<MERS MERS_MINNumber=""/>
<LOAN_FEATURES LienPriorityType="" LoanMaturityDate=""     OriginalPrincipalAndInterestPaymentAmount=""
                ScheduledFirstPaymentDate="">
        <LATE_CHARGE _GracePeriod="" _Rate=""/>
        <NOTE_PAY_TO _City="" _PostalCode="" _State=" " _StreetAddress=""/>
</LOAN_FEATURES>
<BORROWER BorrowerID="" NonPersonEntityIndicator="" _FirstName=""   _HomeTelephoneNumber=""
_LastName=" " _MiddleName="" _SSN=""/>
<PROPERTY _City="" _County="" _PostalCode="" _State="" _StreetAddress="">
        <PARSED_STREET_ADDRESS _DirectionPrefix="" _HouseNumber=""
        _StreetName=" " _StreetType=" "/>
</PROPERTY>
```

This information can be easily populated outside of the document that holds the original eNote in PDF format. This would depend upon the company processes. The above information constitutes data that was originally used to create the final eNote.

Signature Section:
The next important section in the registration request is the signature section which looks like this:

```
<REGISTRY_SIGNATURE>
        <Signature>
                <SignedInfo>
                        <CanonicalizationMethod Algorithm=""/>
                        <SignatureMethod Algorithm=""/>
                        <Reference>
                                <DigestMethod Algorithm=""/>
                                <DigestValue/>
                        </Reference>
                </SignedInfo>
                <SignatureValue/>
        </Signature>
        <Object>
                <SignatureProperties>
                        <SignatureProperty Target="" >
```

```
                    <DateTimeStamp DateTime="" />
                </SignatureProperty>
            </SignatureProperties>
        </Object>
</ REGISTRY_SIGNATURE>
```

This section needs to be populated with the Tamper Evident Signature Value of the PDF eNote. The signature section of the eNote can be extracted from the PDF with a couple of api calls. The most important part of the above section is the Signature Value section which holds the actual value of the signature. The other elements can be standardized for a PDF eNote.

In addition, the signature section must include the datetime stamp of when the Tamper Evident Signature was applied to the eNote.

That is all the information that is need to be able to register an eNote with MERS® eRegistry. The MIN Number and the Signature Value section will be used to uniquely identify the negotiable instrument and to verify its integrity as it moves through it's lifecycle.

The above signature section is geared towards the signature of an XML document in particular, however it will evolve into a more generic format which will be a container for signatures of eDocs of differing formats. Please check with MERS.

Please refer to the MERS® eRegistry Integration Handbooks at www.mersinc.org for further details about the XML sections listed above.

4.15 What are some examples of third-party PDF libraries, and where can I find more information about the PDF community?

Open source third-party PDF library examples:

- http://www.pdfbox.org/
- http://java-source.net/search?q=pdf
- http://www-128.ibm.com/developerworks/opensource/library/os-javapdf/

Commercial third-party PDF library examples:

- http://big.faceless.org/products/pdf/index.jsp
- http://www.adlibsoftware.com/
- http://www.activepdf.com
- http://www.o2sol.com
- http://www.adlibsoftware.com/
- http://www.jawspdf.com/
- http://www.nuance.com/pdfconverter/

- http://www.asppdf.com/
- http://www.cutepdf.com/
- http://www.pdf4net.com
- http://www.pdfonline.com
- http://alt-soft.com/products_xml2pdf.jsp
- http://www.pdftron.com/net/index.html
- http://www.aspose.com

PDF related web portals (third-party PDF products, libraries, tips, opinions, samples, discussions, and other very useful details)

- http://www.pdfzone.com
- http://www.planetpdf.com
- http://www.componentsource.com/relevance/index.html?q=pdf

Please note that the examples can not be considered as recommendation, advice, or endorsement. The examples are referenced for informational purposes only. Each company is responsible for their own due diligence on the subject of the third-party products / libraries.

5. References

a. Government Guides

PDF for U.S. Courts CM/ECF (Case Management/Electronic Case Files)
http://www.uscourts.gov/cmecf/cmecf_about.html

PDF for FDA Electronic Regulatory Submissions and Review
http://www.fda.gov/cder/regulatory/ersr/default.htm#General%20Considerations
http://www.fda.gov/cder/guidance/2867fnl.pdf

United Nations Trade Documents Toolkit
http://unece.unog.ch/etrade/tkhome.aspx
http://unece.unog.ch/etrade/tk3.aspx

b. PDF Specification

The PDF Reference
http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf

XDP - XML Data Package
http://partners.adobe.com/public/developer/en/xml/xdp_2.0.pdf

XFA - XML Forms Architecture
http://partners.adobe.com/public/developer/en/xml/xfapecification_2.2_draft.pdf
Additional XFA materials are also available at:
http://partners.adobe.com/public/developer/xml/index_arch.html

PDF Security Guide
http://www.adobe.com/products/pdfs/AdobePDFSecurityGuide-c.pdf

PDF/A
http://www.aiim.org/article-pr.asp?ID=30413
http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=38920&ICS1=37&ICS2=100&ICS3=99

c. PDF IP Summary

Generally, the IP around PDF is outlined in the PDF Reference (see excerpt below). Additional patent information is found under "Legal Notices for Developers" at:

http://partners.adobe.com/public/developer/support/topic_legal_notices.html

Additionally, throughout the development of PDF-related standards, specific permission has been granted to standards development organizations (AIIM) to freely publish the PDF specification. The goal of this has been to alleviate any concerns around the continued status of the PDF specification in the event of fundamental changes to, or the absence of Adobe Systems, Inc.. The granting of this permission has been instrumental in the development of formal PDF-based ISO standards, and formalizes the long-term availability of the PDF specification.

From the PDF Reference (v1.6):

1.5 Intellectual Property

The general idea of using an interchange format for electronic documents is in the public domain. Anyone is free to devise a set of unique data structures and operators that define an interchange format for electronic documents. However, Adobe Systems Incorporated owns the copyright for the particular data structures and operators and the written specification constituting the interchange format called the Portable Document Format. Thus, these elements of the Portable Document Format may not be copied without Adobe's permission.

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

• *Prepare files whose content conforms to the Portable Document Format*

• *Write drivers and applications that produce output represented in the Portable Document Format*

• *Write software that accepts input in the form of the Portable Document Format and displays, prints, or otherwise interprets the contents*

• *Copy Adobe's copyrighted list of data structures and operators, as well as the example code and PostScript language function definitions in the written specification, to the extent necessary to use the Portable Document Format for the purposes above*

The conditions of such copyright permission are:

• Authors of software that accepts input in the form of the Portable Document Format must make reasonable efforts to ensure that the software they create respects the access permissions and permissions controls listed in Table 3.20 of this specification, to the extent that they are used in any particular document.

These access permissions express the rights that the document's author has granted to users of the document. It is the responsibility of Portable Document Format consumer software to respect the author's intent

• Anyone who uses the copyrighted list of data structures and operators, as stated above, must include an appropriate copyright notice.

This limited right to use the copyrighted list of data structures and operators does not include the right to copy this book, 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).

Acrobat, Acrobat Capture, Adobe Intelligent Document Platform, Adobe Reader, ePaper, the "Get Adobe Reader" Web logo, the "Adobe PDF" Web logo, and all other trademarks, service marks, and logos used by Adobe (the "Marks") are the registered trademarks or trademarks of Adobe Systems Incorporated in the United States and other countries. Nothing in this book is intended to grant you any right or license to use the Marks for any purpose.

6. Definitions

Certification: Certification is a special type of PDF digital signature that allows the author of a document to authenticate it before distribution by signing the layout, data bindings, logic, and scripts inside a document.

Digital Signature: References to Digital Signatures in this document refer to PKI-based signatures unless otherwise stated. PKI digital signatures can be used to definitively establish identity, to detect and/or prevent modifications to documents (tamper-evident sealing), and to establish secure audit trails.

Metadata: Data that describes the characteristics or properties of a document, as opposed to the main contents of a document.

PDF/A: An ISO standard for the long-term preservation of documents. It is based on an earlier version (1.4) of the PDF specification.

Portable Document Format (PDF). PDF is a file format for representing documents in a manner independent of the application software, hardware, and operating system used to create them and of the output device on which they are to be displayed or printed.

XML Data Package (XDP): XDP is a file format that provides an alternate means of expressing a PDF document in a manner where the outer packaging is described with an XML-based syntax rather than a PDF-based syntax.

XML Forms Architecture (XFA): XFA is a template-based grammar and a set of processing rules for the description of interactive forms. XFA is the layout, event, and data template for XML-based PDF.