Visual Document Signatures Profile of the OASIS Digital Signature Services

Working Draft 01, 5 August 2007

Document identifier:

oasis-dss-profiles-visual-document-signatures-wd-01

Technical Committee:

OASIS Digital Signature Services eXtended (DSS-X) TC

Contributors:

Uri Resnitzky, ARX

Related work:

This specification is related to:

- oasis-dss-core-spec-v1.0-os

Abstract:

This document profiles the OASIS DSS core protocols for the purpose of creating and verifying visual document signatures.

Status:

This is a **Working Draft** produced by the OASIS Digital Signature Service Technical Committee. Committee members should send comments on this draft to dss-x@lists.oasis-open.org.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Digital Signature Services eXtended TC web page at http://www.oasis-open.org/committees/dss-x/ipr.php.

Table of Contents

1 Introduction 4
1.1 Terminology 4
1.2 Normative References 4
1.3 Namespaces 4
2 Profile Features 5
2.1 Identifier 5
2.2 Scope 5
2.3 Relationship To Other Profiles 5
2.4 Transport Binding 5
2.5 Security Binding 5
3 Overview (Non-Normative) 6
3.1 Purpose 6
3.2 Motivation 6
3.3 Use cases 6
3.4 Issues Not Covered (out of scope) 6
3.5 New Elements 6
3.6 Operations / Document Lifecycle 6
4 Profile of Signing Protocol 7
4.1 Element <SignRequest> 7
4.1.1 Element <OptionalInputs> 7
4.1.1.1 Optional Input <SignatureType> 7
4.1.1.2 Optional Input <ClaimedIdentity> 7
4.1.1.3 Optional Input <KeySelector> 7
4.1.1.4 Optional Input <SigFieldSettings> 8
4.1.1.5 Optional Input <ReturnPDFTailOnly> 10
4.1.1.6 Optional Input <GraphicImageName> 10
4.1.1.7 Optional Input <SignatureFieldName> 10
4.1.1.8 Optional Input <SignatureReason> 10
4.1.1.9 Optional Input <GMTOffset> 10
4.1.1.10 Optional Input <GraphicImageToSet> 10
4.1.2 Element <InputDocuments> 11
4.2 Element <SignResponse> 11
4.2.1 Element <OptionalOutputs> 11
4.2.1.1 Optional Output <DocumentWithSignature> 11
4.2.1.2 Optional Output <SignedFieldInfo> 11
4.2.1.3 Optional Output <SigFieldSettings> 11
4.2.1.4 Optional Output <AvailableGraphicSignature> 11

4.2.2 Element <SignatureObject> 12
5 Profile of Verifying Protocol 13
5.1 Element <VerifyRequest> 13
5.1.1 Element <OptionalInputs> 13
5.1.1.1 Optional Input <SignatureType> 13
5.1.1.2 Optional Input <UseVerificationTime> 13
5.1.1.3 Optional Input <SignatureFieldName> 13
5.1.2 Element <SignatureObject> 13
5.1.3 Element <InputDocuments> 13
5.2 Element <VerifyResponse> 13
5.2.1 Element <OptionalOutputs> 13
5.2.1.1 Optional Output <FieldsInfo> 13

1 Introduction

The DSS signing and verifying protocols are defined in **[DSSCore]**. As defined in that document, these protocols have a fair degree of flexibility and extensibility. This document profiles these protocols to limit their flexibility and extend them in concrete ways. The resulting profile is suitable for implementation and interoperability.

The following sections describe how to understand the rest of this document.

1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”,

“SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this specification are to be interpreted as described in IETF RFC 2119 **[RFC 2119]**. These keywords are capitalized when used to unambiguously specify requirements over protocol features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

This specification uses the following typographical conventions in text: `<ns:Element>`, `Attribute`, **Datatype**, `OtherCode`.

1.2 Normative References

[Core-XSD]	S. Drees et al. *DSS Schema*. OASIS, February 2007
[DSSCore]	S. Drees et al. *Digital Signature Service Core Protocols and Elements*. OASIS, February 2007
[XML-ns]	T. Bray, D. Hollander, A. Layman. *Namespaces in XML*. http://www.w3.org/TR/1999/REC-xml-names-19990114, W3C Recommendation, January 1999.
[XMLSig]	D. Eastlake et al. *XML-Signature Syntax and Processing*. http://www.w3.org/TR/1999/REC-xml-names-19990114, W3C Recommendation, February 2002.

1.3 Namespaces

The structures described in this specification are contained in the schema file [TBD]. All schema listings in the current document are excerpts from the schema file. In the case of a disagreement between the schema file and this document, the schema file takes precedence.

This schema is associated with the following XML namespace:

`urn:oasis:names:tc:dss:1.0:profiles:VisualDocumentSignatures:schema#`

Conventional XML namespace prefixes are used in this document:

- The prefix `dss:` stands for the DSS core namespace **[Core-XSD]**.

Applications MAY use different namespace prefixes, and MAY use whatever namespace defaulting/scoping conventions they desire, as long as they are compliant with the Namespaces in XML specification **[XML-ns]**.

2 Profile Features

2.1 Identifier

urn:oasis:names:tc:dss:1.0:profiles:VisualDocumentSignatures

2.2 Scope

This document profiles the DSS signing and verifying protocols defined in **[DSSCore]**.

2.3 Relationship To Other Profiles

This profile is based directly on the **[DSSCore]**.

2.4 Transport Binding

This profile is transported using the SOAP 1.2 Transport Binding as defined in the DSS Core spec.

2.5 Security Binding

This profile is secured using the TLS X.509 Server Authentication Binding defined in **[DSSCore]**.

3 Overview (Non-Normative)

3.1 Purpose

The main purpose of this profile is to extend the basic document signing/verification specified in DSS from generic XML and CMS signatures to more specific document formats.

The document formats this profile is intended to handle are primarily those that already include digital signature functionality in their specification, especially those that specify a visual aspect that represents a digital signature.

The currently identified document formats are: PDF [submitted to ISO 32000] which uses CMS signatures, ODF [OASIS ODF 1.2 draft] and OOXML [Ecma 376 submitted to ISO] both which use XML signatures.

3.2 Motivation

The motivation behind this profile is to extend DSS so that it is usable to process popular office / desktop document formats which are in daily use by end users and enterprises.

3.3 Use cases

The envisioned use cases are: creation of a document in existing desktop / server software, submission of the document to be signed by a DSS server, the resulting signed document is capable of being opened, viewed, verified and processed by conforming desktop and server applications. Similarly, existing signed documents [signed by a DSS server or some other conforming software] can be sent to a DSS server for verification.

As such it is clear that interoperability with existing software that processes these document types is of paramount importance in the implementation of this profile.

3.4 Issues Not Covered (out of scope)

Changing or extending existing document format specifications.

3.5 New Elements

The main new data entity this profile introduces is the Signature Field. See Optional Input <SigFieldSettings>. Not all the attributes listed are applicable to all supported document formats.

This profile also adds storage and management of graphical handwritten signature images in the DSS server, perhaps associated with a profile (belonging to a claimed identity) or a specific signing key. The server stored images can be applied when signing a visual signature field.

3.6 Operations / Document Lifecycle

A document can contain any number of signature fields. A signature field can be empty (unsigned) or signed. Therefore the operations in a document lifecycle are: create empty signature field, sign existing signature field (these two can be combined to a single create-and-sign operation), clear a signed signature field, remove a signed or empty signature field. Not all operations are applicable to all document formats and a specific document format may impose certain rules / control on the operations (such as new fields cannot be added once any of the fields is signed).

4 Profile of Signing Protocol

4.1 Element <SignRequest>

4.1.1 Element <OptionalInputs>

No optional inputs other than the ones listed below are supported by this profile.

4.1.1.1 Optional Input <SignatureType>

The <SignatureType> optional input is required and must be one of the following values:

- urn:oasis:names:tc:dss:1.0:core:schema:VisualDocumentSignatures:signature-field-sign - sign an existing signature field.
- urn:oasis:names:tc:dss:1.0:core:schema:VisualDocumentSignatures:signature-field-create-sign - create and sign a new signature field.

The following <SignatureType> values do not actually cause a digital signature to be computed and it is expected that no client authentication is required.

- urn:oasis:names:tc:dss:1.0:core:schema:VisualDocumentSignatures:signature-field-create - create a new signature field.
- urn:oasis:names:tc:dss:1.0:core:schema:VisualDocumentSignatures:signature-field-clear - clear an existing signature field.
- urn:oasis:names:tc:dss:1.0:core:schema:VisualDocumentSignatures:signature-field-remove - remove an existing signature field.

Note that a server MAY impose client authentication requirements before it allows signature fields to be cleared or removed (for example by requiring that only the owner of a certain signing key may be able to remove a signature made by that key).

The following <SignatureType> values do not actually cause a digital signature to be computed, and they do not consume, nor produce a document. However it is expected that client authentication is required.

- urn:oasis:names:tc:dss:1.0:core:schema:VisualDocumentSignatures:enum-graphic-images - enumerate graphical images available for use when signing.
- urn:oasis:names:tc:dss:1.0:core:schema:VisualDocumentSignatures:set-graphic-image - add remove or update a graphical signature image.

4.1.1.2 Optional Input <ClaimedIdentity>

The optional input <ClaimedIdentity> specified in the DSS Core specification is supported by this profile and may be used by the client.

4.1.1.3 Optional Input <KeySelector>

The optional input <KeySelector> specified in the DSS Core specification is supported by this profile and may be used by the client.

4.1.1.4 Optional Input <SigFieldSettings>

This profile introduces a new optional input <SigFieldSettings> which is defined below. This input must be sent by clients when creating a new signature field.

ID

A unique ID assigned to this signature field to be used by references from other elements.

Name

The name of the signature field. The server MAY ignore this attribute when creating a new field, and assign a different field name according to document format specific rules.

Invisible

A flag indicating whether the signature is invisible (true) or visible (false)

Page

The page in the document where the signature field is located. Negative page numbers count from the end of the document backwards.

X

The X coordinate of the signature field rectangle. The coordinate system and units are defined by the document format specification.

Y

The Y coordinate of the signature field rectangle. The coordinate system and units are defined by the document format specification.

Height

The height of the signature field rectangle. The coordinate system and units are defined by the document format specification.

Width

The width of the signature field rectangle. The coordinate system and units are defined by the document format specification.

ShowImage

Indicates whether the signer's graphical signature image should be displayed when the field is signed.

ShowSignerName

Indicates whether the signer's common name should be displayed when the field is signed.

ShowTime

Indicates whether the signature time should be displayed when the field is signed.

ShowReason

Indicates whether the reason should be displayed if available when the field is signed.

EnforceReason

Indicates whether the reason MUST be provided when the field is signed.

ShowLabels

Indicates whether labels (such as "Signed By: ") should be displayed next to the textual parts of the signature field's visual appearance.

ShowLogo

Indicates whether a general logo (such as a corporate logo or signature service logo) should be included in the signature field's visual appearance.

EmptyFieldLabel

The string displayed on an empty (unsigned) signature field rectangle.

Declaration

The declaration made by the signer when agreeing to sign this field.

IntendedSignerName

The common name of the intended signer.

IntendedSignerTitle

The job title / role of the intended signer.

<TimeFormat>

The date and time format of the signing time displayed for the signed field.

<ExtendedInfo>

A mechanism for future enhancements.

```
<xs:complexType name="TimeDateFormatType">
  <xs:attribute name="DisplayAsLocalTime" type="xs:boolean"/>
  <xs:attribute name="DateFormat" type="xs:string"/>
  <xs:attribute name="TimeFormat" type="xs:string"/>
</xs:complexType>

<xs:complexType name="SigFieldSettingsType">
  <xs:sequence>
    <xs:element name="TimeFormat" type="s:TimeDateFormatType"/>
    <xs:element name="ExtendedInfo" minOccurs="0" type="dss:AnyType"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID"/>
  <xs:attribute name="Name" type="xs:string"/>
  <xs:attribute name="Invisible" type="xs:boolean"/>
  <xs:attribute name="Page" type="xs:integer"/>
  <xs:attribute name="X" type="xs:integer"/>
  <xs:attribute name="Y" type="xs:integer"/>
  <xs:attribute name="Height" type="xs:integer"/>
  <xs:attribute name="Width" type="xs:integer"/>
  <xs:attribute name="ShowImage" type="xs:boolean"/>
  <xs:attribute name="ShowSignerName" type="xs:boolean"/>
  <xs:attribute name="ShowTime" type="xs:boolean"/>
  <xs:attribute name="ShowReason" type="xs:boolean"/>
  <xs:attribute name="EnforceReason" type="xs:boolean"/>
  <xs:attribute name="ShowLabels" type="xs:boolean"/>
  <xs:attribute name="ShowLogo" type="xs:boolean"/>
  <xs:attribute name="EmptyFieldLabel" type="xs:string"/>
  <xs:attribute name="Declaration" type="xs:string"/>
  <xs:attribute name="IntendedSignerName" type="xs:string"/>
  <xs:attribute name="IntendedSignerTitle" type="xs:string"/>
</xs:complexType>

<xs:element name="SigFieldSettings" type="s:SigFieldSettingsType"/>
```

4.1.1.5 Optional Input <ReturnPDFTailOnly>

The new optional input <ReturnPDFTailOnly>, which is defined below, modifies the server's behavior when processing PDF files. The PDF specification supports modifications to PDF files to be implemented as a concatenation of the modification data to the end of an existing file. When <ReturnPDFTailOnly> is set to true, the server MUST return only the data to be concatenated to the file instead of the whole file. In this case the data is returned in the <SignatureObject> element instead in the optional output <DocumentWithSignature>. It is expected that usage of this Optional Input will improve performance when processing large PDF files.

```
<xs:element name="ReturnPDFTailOnly" type=”xs:boolean”/>
```

4.1.1.6 Optional Input <GraphicImageName>

The new optional input <GraphicImageName>, which is defined below, allows a client to select the graphic image to use when signing a visible signature field. This input is required in cases where more than one graphical image is available on the server for the signing user.

```
<xs:element name="GraphicImageName" type=”xs:string”/>
```

4.1.1.7 Optional Input <SignatureFieldName>

The new optional input <SignatureFieldName>, which is defined below, allows a client to select the signature field to be processed. This input is required in cases where more than one signature field exists in the document.

```
<xs:element name="SignatureFieldName" type=”xs:string”/>
```

4.1.1.8 Optional Input <SignatureReason>

The new optional input <SignatureReason>, which is defined below, allows a client to specify the reason for signing the document.

```
<xs:element name="SignatureReason" type=”xs:string”/>
```

4.1.1.9 Optional Input <GMTOffset>

The new optional input <GMTOffset>, which is defined below, can be sent by the client as an integer value to represent the client's local time offset in minutes from GMT. This is important in cases where the visual representation of the signed signature field contains the signature time, and the client wishes that string to be displayed in the client's local time.

```
<xs:element name="GMTOffset" type=”xs:integer”/>
```

4.1.1.10 Optional Input <GraphicImageToSet>

The new optional input <GraphicImageToSet>, which is defined below, must be sent by the client whenever <SignatureType> is set-graphic-image. Multiple occurrences of this optional input are allowed. If a graphical signature image with the same GraphicImageName attribute already exists, it will be updated, unless the <GraphicImage> element is empty, in which case it will be deleted. If an image with the given name does not exist a new image will be created. Note that the format of the image (JPEG, PNG, etc) is represented using the MimeType attribute of the <GraphicImage> element.

```
<xs:complexType name="GraphicImageType">
  <xs:sequence>
```

```
    <xs:element name="GraphicImage" type="dss:Base64Data"/>
  </xs:sequence>
  <xs:attribute name="GraphicImageName" type="xs:string"/>
</xs:complexType>

<xs:element name="GraphicImageToSet" type="s:GraphicImageType"/>
```

4.1.2 Element <InputDocuments>

The client MUST NOT send <DocumentHash> input documents.

The client MUST send <Document> input documents with the <Base64Data> element set to the document content and the MimeType attribute set to the file type. (for example: “application/pdf” for PDF files).

4.2 Element <SignResponse>

4.2.1 Element <OptionalOutputs>

4.2.1.1 Optional Output <DocumentWithSignature>

The optional output <DocumentWithSignature> is always returned by the server in cases involving signature fields unless the optional input <ReturnPDFTailOnly> is set to true. In this last case <SignatureObject> will be retuned instead.

The <DocumentWithSignature> element will always contain a <Base64Data> element containing the output file whose MimeType attribute will be set according to the file type.

4.2.1.2 Optional Output <SignedFieldInfo>

This profile introduces a new optional output <SignedFieldInfo> which is defined below. This output will be returned when the optional input <SignatureType> is set to signature-field-sign.

```
<xs:complexType name="SignedFieldInfoType">
  <xs:sequence>
    <xs:element name="Certificate" minOccurs="0" type="xs:base64Binary"/>
    <xs:element name="GraphicImage" minOccurs="0" type="s:GraphicImageType"/>
  </xs:sequence>
  <xs:attribute name="SignerName" type="xs:string"/>
  <xs:attribute name="IsSigned" type="xs:boolean"/>
  <xs:attribute name="SignatureTime" type="xs:dateTime"/>
  <xs:attribute name="Reason" type="xs:string"/>
</xs:complexType>

<xs:element name="SignedFieldInfo" type="s:SignedFieldInfoType"/>
```

4.2.1.3 Optional Output <SigFieldSettings>

The previously defined optional input <SigFieldSettings> may also be retuned by the server as an optional output. This output will be returned when the optional input <SignatureType> is set to signature-field-create. The main purpose of this is to let the client know the name of the newly created signature field in cases where the server assigns it.

4.2.1.4 Optional Output <AvailableGraphicSignature>

This profile introduces a new optional output <AvailableGraphicSignature> which is defined below. This output will be returned when the optional input <SignatureType> is enum-graphic-

images. Multiple occurrences of this optional output may be returned by the server in cases where more than one graphical signature image is available.

```
<xs:element name="AvailableGraphicSignature" type="s:GraphicImageType"/>
```

4.2.2 Element <SignatureObject>

The <SignatureObject> element is returned by the server only in case the optional input <ReturnPDFTailOnly> is set to true.

The <SignatureObject> element will always contain a <Base64Signature> element containing the output signature whose Type attribute will be set to the value of the <SignatureType> optional input.

5 Profile of Verifying Protocol

5.1 Element <VerifyRequest>

5.1.1 Element <OptionalInputs>

No optional inputs other than the ones listed below are supported by this profile.

5.1.1.1 Optional Input <SignatureType>

The <SignatureType> optional input is required and must be one of the following values:

- urn:oasis:names:tc:dss:1.0:core:schema:VisualDocumentSignatures:signature-field-verify - verify existing signature fields.

5.1.1.2 Optional Input <UseVerificationTime>

The optional input <UseVerificationTime> specified in the DSS Core specification is supported by this profile and may be used by the client.

5.1.1.3 Optional Input <SignatureFieldName>

The previously defined optional input <SignatureFieldName> allows a client to select the signature field to be processed. When this input is omitted and more than one signature field exists in the document, all the fields in the document will be processed.

5.1.2 Element <SignatureObject>

The <SignatureObject> element MUST NOT be sent by the client.

5.1.3 Element <InputDocuments>

Please refer to the section describing the <InputDocuments> elements in the profile of the signing protocol above.

5.2 Element <VerifyResponse>

5.2.1 Element <OptionalOutputs>

5.2.1.1 Optional Output <FieldsInfo>

The new optional output element <FieldsInfo> defined below MUST be retuned by the server as an optional output. Multiple occurrences of this element will be returned, one for each signature field in the input document. The <FieldStatus> subelement will only be retuned for signed (non-empty) signature fields.

```
<xs:complexType name="VerificationStatusType">
  <xs:attribute name="SignatureValid" type="xs:boolean"/>
  <xs:attribute name="CertificateValid" type="xs:boolean"/>
</xs:complexType>

<xs:element name="FieldsInfo">
```

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SigFieldSettings" type="s:SigFieldSettingsType"/>
      <xs:element name="SignedFieldInfo" type="s:SignedFieldInfoType"/>
      <xs:element name="FieldStatus" type="s:VerificationStatusType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```