# Asynchronous Processing Abstract Profile of the OASIS Digital Signature Services

## Committee Draft, 24 December, 2004 (Working Draft 05)

**Document identifier:**
oasis-dss-1.0-profiles-asynchronous-processing-spec-cd-01

**Location:**
http://docs.oasis-open.org/dss/

**Editor:**
Andreas Kuehne, *individual* <kuehne@klup.de>

**Contributors:**
Trevor Perrin, *individual* <trevp@trevp.net>

Pieter Kasselman, *Betrusted* <pkasselman@betrusted.com>

**Abstract:**
This draft profiles the OASIS DSS core protocol for asynchronous processing. This profile is intended to be generic, so it may be combined with other profiles freely.
The protocol is designed to be similar to the asynchronous aspects of the XML Key Management Specification **[XKMS]**.

**Status:**
This is a **Committee Draft** produced by the OASIS Digital Signature Service Technical Committee. Committee members should send comments on this draft to dss@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 Service TC web page at http://www.oasis-open.org/committees/dss/ipr.php.

# Table of Contents

# 1 Introduction

This is an *abstract profile*. Further profiles will build on this one to provide a basis for implementation and interoperability.

This draft profiles the OASIS DSS core protocol for asynchronous processing. Although most applications of the OASIS Digital Signature Service supply the results immediately there is a demand for deferred delivering of results. E.g. the German Signature Law explicitly requires the commitment of the certificate holder or at least a time slot for the certificate holder to deny the signing request **[SigG]**.

Another use case for a asynchronous protocol may arise in a verification request if a minimum latency between creation and verification has to be respected.

This profile is intended to be generic, so it may be combined with other profiles freely.
A protocol for asynchronous processing is already defined in the XML Key Management Specification **[XKMS]**. This profile borrows ideas from the XKMS protocol for the OASIS Digital Signature Service.

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

## 1.1 Notation

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 Namespaces

The structures described in this specification are contained in the schema file **[XYZ-XSD]**. 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:asynchronousprocessing:1.0
```

If a future version of this specification is needed, it will use a different namespace.

Conventional XML namespace prefixes are used in this document:

- The prefix `async:` stands for this profiles namespace **[Core-XSD]**.
- The prefix `dss:` (or no prefix) stands for the DSS core namespace **[Core-XSD]**.
- The prefix `ds:` stands for the W3C XML Signature namespace **[XMLSig]**.

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]**.

## 1.3 Overview (Non-normative)

This profile defines a simple mechanism for asynchronous signing and verification requests. This profile is similar to the asynchronous processing protocol defined in the XKMS spec **[XKMS]**.

In the first call the client supplies its input values as defined in the core and the applied profiles. The client MUST supply the RequestId attribute to correlate subsequent calls.

106 The server may reply synchronously with the appropriate result.

107 On the other hand the server may reply with an ‘empty’ result, giving the `<ResultMajor>` code
108 ‘Pending’. In this case the client may initiate a <PendingRequest> call from time to time with the
109 `RequestId` of the initial call included in the `OriginalRequestId` attribute.

110 When the server finally succeeds with its processing the results will be delivered to the client at its next
111 polling call. In this case the `<ResultMajor>` must not be `‘Pending’` but the `<ResultMajor>`
112 resulting from the request processing.

113

114 A notification mechanism isn’t defined yet, but may be subject to following versions of this profile.

115

# 2 Profile Features

## 2.1 Identifier

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

Add an `<AdditionalProfile>` element containing this URI to use this profile.

## 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]**.

This profile is an abstract profile which is not implementable directly.

This profile is intended to be combined with other profiles freely.

## 2.4 Signature Object

This profile does not specify or constrain the type of signature object.

## 2.5 Transport Binding

This profile does not specify or constrain the transport binding.

## 2.6 Security Binding

This profile does not specify or constrain the security binding.

# 3 Polling Protocol

The polling protocol extends the core protocol using the <PendingRequest> element for initiating a polling request. This is different from the initial request because the request specific data was already transmitted.

## 3.1 Element <PendingRequest>

`RequestID`

This attribute is used to correlate requests with responses. When present in a request, the server MUST return it in the response.

`OriginalRequestID`

This attribute is used to correlate initial requests with polling requests. The client MUST supply it with each polling request. Its value must match the RequestID sent in an initial <SignRequest> or <VerifyRequest> message (see sections 4.1 and 5.1).

The `<PendingRequest>` element doesn't define any child elements.

```
<xs:element name="PendingRequest">
<xs:complexType name="PendingRequestType">
      <xs:attribute name="RequestID" type="xs:string" use="optional"/>
      <xs:attribute name="OriginalRequestID" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
```

# 4 Profile of Signing Protocol

## 4.1 Element <SignRequest>

`RequestID`

This attribute is used to correlate requests with responses. [DSSCore] defines this attribute as optional. This profile makes the attribute mandatory: the client MUST supply it with the initial request.

### 4.1.1 Element <OptionalInputs>

This profile doesn't interfere with the optional inputs from **[DSSCore]** or from other profiles.

### 4.1.2 Element <InputDocuments>

This profile doesn't interfere with the optional inputs from **[DSSCore]**.

## 4.2 Element <SignResponse>

### 4.2.1 Element <Result>

This profile defines the additional <`ResultMinor`> codes, which MAY be returned in conjunction with a <ResultMajor> code of RequesterError to provide more detailed error reporting:

`urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultminor:RequestIdInUse`

This result value means that an operation with the given RequestId is still in progress. The client may retry with a different RequestId. This result code shows up only in response to a <SignRequest>.

`urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultminor:RequestIdRequired`

This result value means that this operation requires a RequestId for this Request.

`urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultminor:RequestIdUnknown`

This result value means that an operation included a RequestId that the server didn't recognize. This result code shows up only in response to a <PendingRequest>.

### 4.2.2 Element <ResultMajor>

This profile defines the additional <`ResultMajor`> code, which may show up in response to a <SignRequest> or <PendingRequest>:

`urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending`

This result value means that the operation did not finish yet. Subsequent requests may return this result code again. After the server has finished the operation the call will return the signing result indicated by the `urn:oasis:names:tc:dss:1.0:resultmajor:Success` value or an error code.

### 4.2.3 Element <OptionalOutputs>

This profile doesn't interfere with the optional outputs from **[DSSCore]**. No new optional outputs are defined by this profile.

If the server returns the `<ResultMajor>` code

`urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending`

the contents of the `<OptionalOutputs>` element is undefined.

If the server returns the `<ResultMajor>` code

`urn:oasis:names:tc:dss:1.0:resultmajor:Success`

the `<OptionalOutputs>` MUST contain the results defined by the accompanying profiles as expected in synchronous operation.

### 4.2.4 Element <SignatureObject>

If the server returns the `<ResultMajor>` code

`urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending`

the content of the `<SignatureObject>` element is undefined.

If the server returns the `<ResultMajor>` code

`urn:oasis:names:tc:dss:1.0:resultmajor:Success`

the `<SignatureObject>` MUST contain the results defined by the accompanying profiles as expected in synchronous operation.

# 5 Profile of Verifying Protocol

## 5.1 Element <VerifyRequest>

`RequestID`

This attribute is used to correlate requests with responses. The client MUST supply it with the initial request.

### 5.1.1 Element <OptionalInputs>

This profile doesn't interfere with the optional inputs from **[DSSCore]** or from other profiles.

### 5.1.2 Element <SignatureObject>

This profile doesn't interfere with the element defined from **[DSSCore]**.

### 5.1.3 Element <InputDocuments>

This profile doesn't interfere with the element defined from **[DSSCore]**.

## 5.2 Element <VerifyResponse>

### 5.2.1 Element <Result>

This profile defines the additional <`ResultMinor`> code

`urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultminor:RequestIdInUse`

. This result value defines that an operation with the given RequestId did not finish yet. The client may retry with a different RequestId.

This profile defines the additional <`ResultMinor`> code

`urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultminor:RequestIdRequired`

. This result value defines that this operation requires a RequestId for this Request. The client retry with the RequestId defined.

### 5.2.2 Element <ResultMajor>

This profile defines the additional <`ResultMajor`> code

`urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending`

. This result value defines that the operation did not finish yet. Subsequent requests will return the signing result indicated by the `urn:oasis:names:tc:dss:1.0:resultmajor:Success` value or an error code.

### 5.2.3 Element <OptionalOutputs>

This profile doesn't interfere with the optional outputs from **[DSSCore]**. No new optional outputs are defined by this profile.

If the server returns the <ResultMajor> code

urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending

the content of the <OptionalOutputs> element is undefined.

If the server returns the <ResultMajor> code

urn:oasis:names:tc:dss:1.0:resultmajor:Success

the <OptionalOutputs> MUST contain the results defined by the accompanying profiles the way are expected in synchronous operation.

# 6 References

## 6.1 Normative


**[Core-XSD]** T. Perrin et al. *DSS Schema*. OASIS, **(MONTH/YEAR TBD)**

**[DSSCore]** T. Perrin et al. *Digital Signature Service Core Protocols and Elements*. OASIS, **(MONTH/YEAR TBD)**

**[RFC 2119]** S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. IETF RFC 2396, August 1998.

http://www.ietf.org/rfc/rfc2396.txt.

**[XML-ns]** T. Bray, D. Hollander, A. Layman. *Namespaces in XML.* W3C Recommendation, January 1999.

http://www.w3.org/TR/1999/REC-xml-names-19990114

**[XMLSig]** D. Eastlake et al. *XML-Signature Syntax and Processing.* W3C Recommendation, February 2002.

http://www.w3.org/TR/1999/REC-xml-names-19990114

**[SigG]** Framework for Electronic Signatures, Amendment of Further Regulations Act (Signaturgesetz – SigG), 21 May 2001. http://www.regtp.de/imperia/md/content/tech_reg_t/digisign/119.pdf

**[XKMS2]** Phillip Hallam-Baker *XML Key Management* Specification (XKMS 2.0) W3C Candidate Recommendation, 5 April 2004.
http://www.w3.org/TR/2004/CR-xkms2-20040405/

# Appendix A. Revision History

| Rev | Date | By Whom | What |
|---|---|---|---|
| wd-01 | 2004-04-17 | Andreas Kuehne | Initial version |
| wd-02 | 2004-05-09 | Andreas Kuehne | Modifying the profile for ‘PendingRequest’ |
| wd-03 | 2004-06-28 | Andreas Kuehne | Correlation of initial and subsequent calls optimized |
| wd-04 | 2004-08-21 | Andreas Kuehne | Added additional return codes. Schema snippets inserted. |
| Wd-05 | 2004-11-24 | Andreas Kuehne | ResponseMechanism deferred to a later version, no real need now |

# Appendix B. Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.