

XML Signature Syntax & Processing

W3C XML Security Specifications
Maintenance WG, 2007-05-02

Thomas Roessler <tlr@w3.org>

```
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
    (<Reference URI? >
      (<Transforms>)?
      <DigestMethod>
      <DigestValue>
      </Reference>)+
  </SignedInfo>
  <SignatureValue>
  (<KeyInfo>)?
  (<Object ID?>)*
</Signature>
```

Essentials

- `<SignedInfo/>` is the material that's *really* signed.
- Includes information about:
 - Algorithms
 - CanonicalizationMethod
 - SignatureMethod
 - Other material covered by the signature (`<Reference>` etc)
 - Hash values

```
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
    (<Reference URI? >
      (<Transforms>)?
      <DigestMethod>
      <DigestValue>
      </Reference>)+
  </SignedInfo>
  <SignatureValue>
  (<KeyInfo>)?
  (<Object ID?>)*
</Signature>
```

<CanonicalizationMethod/>

- Required element
- Identifies method used to canonicalize the <SignedInfo> element before hashing.
- Does not automatically apply to referenced material.

<SignatureMethod/>

- e.g., dsa-sha1
`http://www.w3.org/.....#dsa-sha1`
- “the algorithm that is used to convert the canonicalized `SignedInfo` into the `SignatureValue`”

<Reference>

- Includes the
 - digest method (<DigestMethod>)
 - resulting digest value (<DigestValue>)
 - calculated over the *identified data object*
- MAY include transformations that produced the input to the digest value.

```
<Transforms>
```

```
  <Transform Algorithm="..." />
```

```
</Transforms>
```

Reference processing model

The data-type of the result of URI dereferencing or subsequent Transforms is either an octet stream or an XPath node-set.

The Transforms specified in this document are defined with respect to the input they require. The following is the default signature application behavior:

- If the data object is an octet stream and the next transform requires a node-set, the signature application **MUST** attempt to parse the octets yielding the required node-set via **[XML]** well-formed processing.
- **If the data object is a node-set and the next transform requires octets, the signature application MUST attempt to convert the node-set to an octet stream using Canonical XML [XML-C14N].**

Proposed Change to Reference Processing Model

- When generating a signature, add an explicit `<ds:Transform>` to encode the canonicalization algorithm that is to be used to convert a node-set into an octet string.
- If the last transformation that is chosen by the application generates a node-set, append C14N 1.1 to the list of transforms.
- Penalize current practice on receiver's side?

Proposed Change to Mandatory Algorithms

- Use Canonical XML 1.1 Instead of Canonical XML 1.0
- That's search & replace throughout the spec

Steps Ahead

- Last Call
- Develop text for PER
- Develop test cases for interop testing
- Interop testing coordinated with CR phase for C14N 1.1 in XML Core
- PR for C14N 1.1 (XML Core) synchronized with PER for xmldsig-core

