C14n 1.1

Canonical XML 1.1

Konrad.Lanz@iaik.tugraz.at

Konrad Lanz

- Digital Signature Services OASIS-DSS
  - IAIK (Inst. f. angew. Informationsverarbeitung und Kommunikation)
  - SIC
    - Stiftung Secure Information and Communication Technology
  - TUG (Technische Universität Graz)
- OASIS-DSS TC Voting Member
- W3C
  - Zentrum für Sichere Informationstechnologie (A-SIT)
  - W3C XML CORE Working Group
    - Canonicalization (c14n)

XML Normal-Form (Canonicalization)

- implicit
  - next input OctetStreamData
  - or Digest
- explicit
  - <ds:Transform>
  - <ds:SignedInfo>
    - <ds:CanonicalizationMethod>

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

canonicalize documents or NodeSets

- canonicalize
  - whole document
  - subset of the document's nodes
    - XPointer to dereference only parts of a document
    - XPath Filter and XPath Filter 2.0 transforms

known issues

- xml:base
  - special values of xml:base
  - inheriting xml:base values
- xml:id
- implicit use of C14n 1.0 by XML Signature
- Further considerations for C14N/1.1
  - xml:base and URI reference simplification
  - XML infoset strategy for canonicalizing XML base

xml:base special values

- xml:base values may
  - consist of only a fragment identifier (no-op)
    - xml:base="#some-fragment"
  - be empty (no-op)
    - xml:base=""
  - be absolute or relative URI references

xml:base inheritance

- relative URI references in xml:base attribute
  - depend on
    - chain of xml:base along element's ancestor axis
    - base URI of the document entity or external entity containing the element.

xml:id non-inheritance

- C14N/1.0 cannot be applied to documents containing xml:id attributes.

- Inheritance of any xml:id attributes would produce a wrong or a badly-formed document.

Diff (C14n , C14n 1.1)

- Diff
  - Section 2.4 Document Subsets
    - [Definition:] **Simple inheritable attributes**
    - xml:id attribute is not a simple inheritable attribute
    - xml:base fix up

C14n 1.1 has to be used EXPLICITLY

- data object level
  - an explicit C14n 1.1 <ds:Transform>
    - before each <ds:Transform>
      - requires an octet stream as input, but is applied to a node-set
  - if the last transform returns a note-set
    - append an explicit C14n 1.1 <ds:Transform> as the last <ds:Transform> before the digest input.
- <ds:SignedInfo> level use this URI inside <ds:CanonicalizationMethod>

Conclusion

+ Compatible

- Complex to be used
- Increased size

→ Future: <ds:CanonicalizationMethod>
specify the implicit (default)
node-set to octet stream conversion

That's is for C14n 1.1

- Thanks for your Attention !
- However there are issues remaining …

References

- **Canonical XML 1.1**
- **Using XML Digital Signatures in the 2006 X**
- **Known Issues with Canonical XML 1.0 (C14**

C14, C14N 1.1
XML 1.1/ Namespaces 1.1

- first look -> no reason that C14N 1.1 couldn't be used with XML 1.1
- second look -> XPath 1.0 data model for an XML 1.1 document not defined
  - NS 1.1 allows the undeclaring of a namespace prefix
  - undefined how XPath 1.0 would treat this.
  - analogy to xmlns=""

C14n
NodeSetData Input

Intuitive view:

- If a namespace is declared in the input, then it must be declared in the output (iff used).
  - usual case
    - XPath data model all elements bear their nsdecls in scope
    - distributed to all elements along the descendant-or-self axis

Current processing:

- Exception: all the namespace nodes (NsDecls) along an element's (E) ancestor axis declaring E's namespace (N) are removed from the C14n input nodeset.

- violate the namespace constraint: "Prefix Declared"
  - maybe still be valid in some surrounding context

Continued …

- already be problems with C14N and NS 1.0
  - not preserving prefixes in some cases
  - Currently not a problem, no requirement in C14n to return well formed namespace conformant XML under all circumstances
  - maybe a potential new requirements for c14n for future canonical XML specifications.
  - alternative view
    - fixup is necessary in C14n
      - prevent the creation of output violating the namespace constraint: "Prefix Declared"

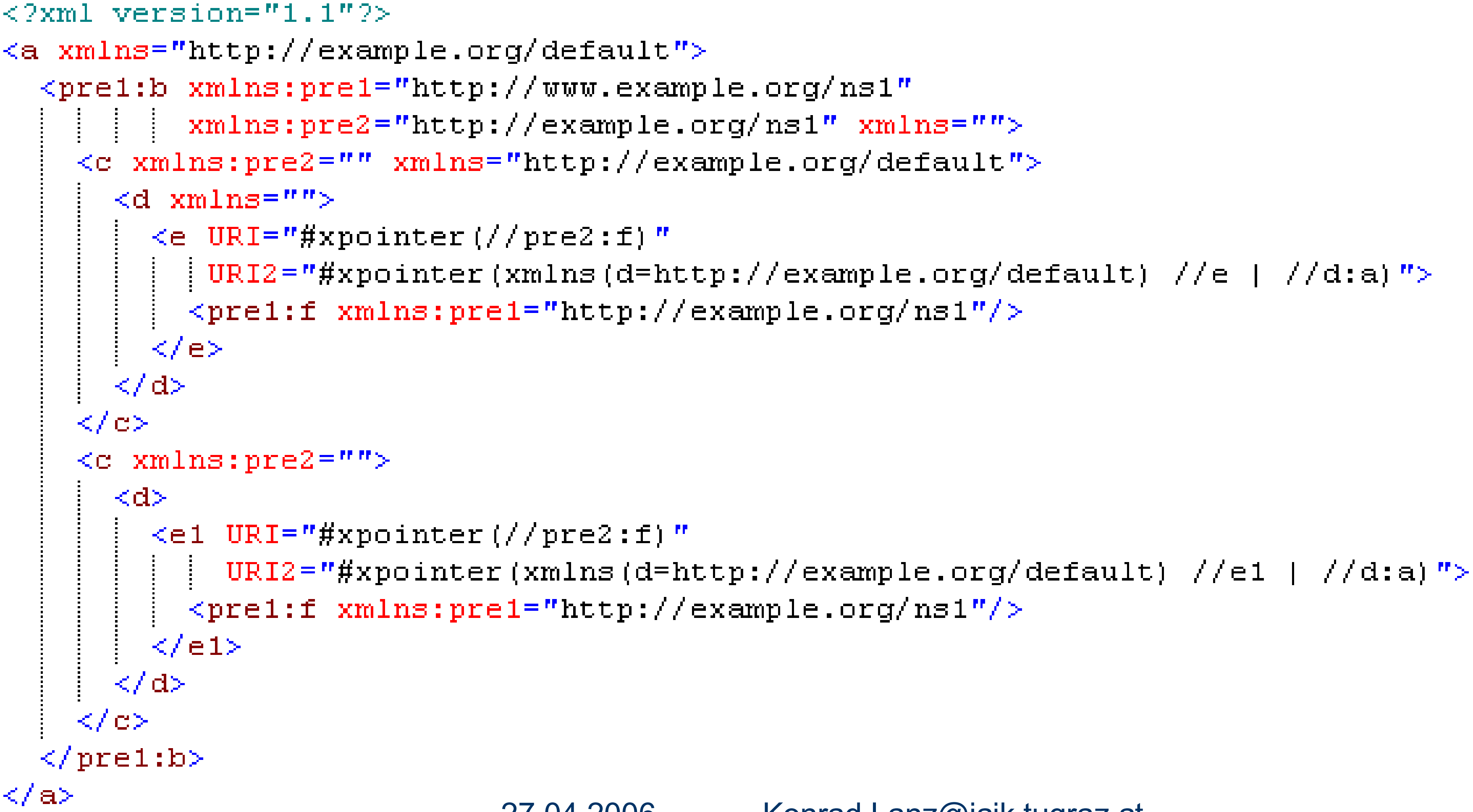
Solution1: Changing XPath 1.0

- second bullet in section 5.4 "Namespace Nodes"
  - This means that an element will have a namespace node:
    - for every first attribute **(nearest per prefix)** on an ancestor element whose name starts with "xmlns:" **having a non-empty value** unless the element itself or a nearer ancestor redeclares the prefix **with a non-empty value;**
      Note: empty values appear in XML 1.1 to undeclare a namespace prefix.
- maintain "undeclarations" ?
  - Further changes to c14n specifications required

Solution2: undeclaration = redeclaration …

- treat xmlns:prefix="" like a redeclaration
  - that simply "overwrite" a prefix with a "non-namespace"
  - c14n specifications could remain mostly untouched.

C14n & XML 1.1 Example



```xml
<?xml version="1.1"?>
<a xmlns="http://example.org/default">
  <pre1:b xmlns:pre1="http://www.example.org/ns1"
          xmlns:pre2="http://example.org/ns1" xmlns="">
    <c xmlns:pre2="" xmlns="http://example.org/default">
      <d xmlns="">
        <e URI="#xpointer(//pre2:f)"
           URI2="#xpointer(xmlns(d=http://example.org/default) //e | //d:a)">
          <pre1:f xmlns:pre1="http://example.org/ns1"/>
        </e>
      </d>
    </c>
    <c xmlns:pre2="">
      <d>
        <e1 URI="#xpointer(//pre2:f)"
            URI2="#xpointer(xmlns(d=http://example.org/default) //e1 | //d:a)">
          <pre1:f xmlns:pre1="http://example.org/ns1"/>
        </e1>
      </d>
    </c>
  </pre1:b>
</a>
```

C14n & XML 1.1 Example

```
a - xmlns="http://example.org/default"
|
+ - pre1:b - xmlns="" - ( missing )
      |    - xmlns:pre1="http://example.org/ns1"
      |    - xmlns:pre2="http://example.org/ns1"
      |
      + - e - URI="#xpointer(//pre1:e)"
      |   | - URI2="#xpointer(...)"
      |   | - xmlns="" - ( missing )
      |   | - xmlns:pre1="http://example.org/ns1"
      |   | - xmlns:pre2="" - ( or missing )
      |   |
      |   + - pre1:f - xmlns:pre1="http://example.org/ns1"
      |              - xmlns="" - ( missing )
      |              - xmlns:pre2="" - ( or missing )
      |
      + - e1- URI="#xpointer(//pre1:e)"
          | - URI2="#xpointer(...)"
          | - xmlns="" - ( missing )
          | - xmlns:pre1="http://example.org/ns1"
          | - (xmlns:pre2 is not in the node set)
          |
          + - pre1:f - xmlns="" - ( missing )
                     - xmlns:pre1="http://example.org/ns1"
                     - (xmlns:pre2 is not in the node set)
```

References

- Secure XML
  - Donald E. Eastlake III and Kitty Niles, Addison Wesley, 2003
- XMLDSig
  - http://www.w3.org/TR/xmldsig-core/
  - http://www.ietf.org/rfc/rfc4051.txt
- Canonicalization
  - http://www.w3.org/TR/xml-exc-c14n/
- XMLEnc
  - http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

JAVA

- XML-DSig (JSR 105)
  - http://www.jcp.org/en/jsr/detail?id=105
- XML-Enc (JSR 106)
  - http://www.jcp.org/en/jsr/detail?id=106

Thanks !
SIC – XSect Toolkit

- IAIK XML Signature Library (IXSIL) Nachfolger
- Java XML Digital Signatures APIs (JSR105)
- Java XML Digtial Encryption APIs (JSR106)

- http://www.sic.st
- http://jce.iaik.tugraz.at/sic/products/xml_security

- Thanks for your Attention.