Canonical XML

Consistent. Predictable. W3C Canonical XML in PostgreSQL

Jim Jones
jim.jones@uni-muenster.de
University of Münster

Introduction

XML is widely used in data exchange, signatures, and interoperability workflows — but comparing XML documents is notoriously unreliable due to differences in formatting, whitespace, attribute order, or namespace declarations.

For example, the following XML documents are logically equivalent but not byte-equal:

```
<foo>
  <bar a2="1" a1="0"/>
</foo>
```

vs.

```
<foo><bar a1=’0’  a2=’1’></bar></foo>
```

Despite having the same semantic structure, their differing attribute order, quoting style, and whitespace make them unequal for hashing, digital signatures, or diffing.

The W3C Canonical XML standard (C14N) solves this by defining a strict, consistent serialization of XML documents.

Patch Overview

PostgreSQL currently lacks a built-in way to produce such canonical XML, which makes XML-based digital signatures, hashing, and reliable diffing quite challenging.

This patch introduces `xmlcanonicalize()`, a native PostgreSQL function that outputs canonical XML directly from SQL, following W3C standards.

```
xmlcanonicalize(doc xml [, keep_comments boolean
DEFAULT true) → xml
```

`doc`: the XML document to be canonicalized.

`keep_comments`: flag to control whether XML comments from the input document are preserved or discarded. If omitted, it defaults to `true`.

XMLCanonicalize in a nutshell:

- Produces W3C Canonical XML (C14N) documents.
- Based on libxml2's canonicalization API.
- Supports comments via optional parameters.

Examples

Preserving XML comments:

```
SELECT xmlcanonicalize(
  '<foo>
    <!-- comment -->
    <bar a2="1" a1="0"/>
  </foo>'
);
                   xmlcanonicalize
-------------------------------------------------------
 <foo><!-- comment --><bar a1="0" a2="1"></bar></foo>
```

Discarding XML Comments:

```
SELECT xmlcanonicalize(
  '<foo>
    <!-- comment -->
    <bar a2="1" a1="0"/>
  </foo>',
  false
);
           xmlcanonicalize
---------------------------------------
 <foo><bar a1="0" a2="1"></bar></foo>
```

Collaboration & Feedback

This patch has received valuable reviews already, but further input is welcome. We're especially interested in:

- Naming conventions that align with PostgreSQL’s style
- Suggestions for making the function intuitive and discoverable
- Experience with other canonicalization tools?

Your feedback can help shape a user-friendly and robust addition to PostgreSQL's XML toolbox.

Commitfest item:

Universität Münster

PGConf.dev 2025