EXIP
Embeddable EXI implementation in C

Rumen Kyusakov
PhD student, EISLAB – Luleå University of Technology
Supervisor: Jens Eliasson

Grenoble France, March 10, 2011
“EXI is a very compact representation for the Extensible Markup Language (XML) Information Set that is intended to simultaneously optimize performance and the utilization of computational resources.”

Efficient XML Interchange (EXI) Format 1.0
W3C Proposed Recommendation 20 January 2011
Outline

• Project goals
• Design principles
  • API
  • String tables
  • EXI grammars
  • Schema representation
• Structure
• Project stage
• Milestones
• Why to participate?
• Conclusion
Project goals

➢ Efficiency
  • Small code size
  • Low memory usage
  • Fast execution

➢ Portability
  • Limited external dependencies
  • Preserving its features across different platforms

➢ Modularity
  • Adding or removing features
  • Extendability
Design principles – overview

XML Information Set

- Plain XML
- Binary EXI

EXIP

- Content IO
- Schema Parser
- Grammars
- String Tables
- Stream IO
- Common Utils

<EXIP API>
EXIP API

• Current EXI implementations use SAX, DOM and StAX
  ✔ Easy to migrate legacy applications using XML to EXI
  ✔ Easy to reuse the knowledge of XML developers
  ✗ Do not support typed data
  ✗ Inefficient when applied to EXI

• EXIP API
  • Use streaming – similar to SAX when parsing and to StAX when serializing
  • Typed
  • Low-level
  • Intuitive
String tables in EXIP

- Strings in EXIP are length prefixed
- Support for different representations can be easily added
Schema representation
Project structure

- Testing with Check Unit Testing Framework
- Automated builds with Make or MS Visual Studio
  → Platforms tested: Linux 32/64 bits, Windows 7, Mulle 16 bits
- Support for tracing (logging debug messages)
- Doxygen documentation
Project stage

✔ Adequate and fairly stable API
✔ Stable project structure

✗ Only default processing options in the EXI header are supported
✗ Compression not implemented
✗ Schema processing in a very early stage – only very very simple documents
Milestones

- November 23, 2010 – First pre-alpha release
- March 2011 – Bugfix pre-alpha release
  - Improved API
  - Better unit-test coverage
  - Much more stable default options processing
  - Very simple and unstable schema-enabled parsing and serialization
- September 2011 – Alpha release
  - Stable default options processing
  - Unstable but full-featured schema processing
  - Partial support for EXI compression
- 2012 – Beta release
Why to participate?

 Benefits from the open source development community:
  - Source code
  - Testing efforts
  - Users feedback

 Influence the priority of features, bug fixes etc.
 Up-to-date information on project maturity, release dates and known issues
Conclusion

• Currently EXIP:
  • has stable API and project structure
  • is not mature enough for real-world usage
  • is written in highly portable manner

• Project statistics:
  • 77 commits
  • 164 downloads