

# Presentation

## Extensible Binary Universal Protocol

Work Release 16  
Project Overview

XBUP Project  
<http://xbup.sf.net/>

# Introduction

XBUP should be universal binary protocol for communication and long-term data storage purposes with unified tree structure.

- Open and free of charges
- Extensible and with compatibility support
- With formally proved design
- Universal
- Independent

# Motivation

Why new format/protocol?

- Unified Access
- New features

Why binary?

- Better processing performance
- More effective bandwidth and memory utilization
- Language independent
- Better abstraction and verifiability

# Similar Projects

- XML (eXtensible Markup Language)  
Activities around Efficient/Binary XML (about 50 projects): EXI, BXML, Fast Infoset, WBXML, XBIS, ...
- ASN.1 (Abstract Syntax Notation One)  
International standard in Telephone Industry for binary data interchange.
- Others  
Nasa HDF5, EBML (Matroska), RIFF, Postscript, ...

# Low Level Goals

- Binary, spoken language independent
- Unlimited number values and data sizes
- Wide extensible
- Multiple formats combinations
- Random access acceleration
- Direct streaming support

# Middle Level Goals

- Block types, block catalogs, document scheme, hierarchy and inheritance
- General editing tools
- Compression and encryption support
- Cyclic redundancy checks, correction blocks, block fragmentation
- Document range limitation
- Remote procedure calls

# High Level Goals

- Auto conversion to older versions
- Plugins for viewing and other operations
- Operating systems integration
- Programming languages integration, multiple paradigm support
- Script language and processing instruction support
- Hierarchical self-descriptiveness

# Principles

Following principles will be mentioned in next part of protocol description:

- Unlimited Numeric Binary Encoding
- Simple Block-Tree Structure
- Hierarchical Types
- Specification Catalog
- Transformations
- Multilevel Architecture

# Unlimited Numeric Binary Encoding

Numbers are stored in UTF-8 similar form, first unary length and then binary value with recursive application. Code is prefix and deterministic.

```
0xxxxxxx = 0..7Fh
10xxxxxx xxxxxxxx = 0..3FFFh
110xxxxx xxxxxxxx xxxxxxxx = 0..1FFFFFh
1110xxxx xxxxxxxx xxxxxxxx xxxxxxxx = 0..0FFFFFFFh
...
11111110 xxxxxxxx .. xxxxxxxx = 0..0FFFFFFFFFFFFFFFFFh
      \_____ 7 times ____/
11111111 00000000 xxxxxxxx .. xxxxxxxx = 0..0FFFFFFFFFFFFFFFFFh
      \added_/ \_____ 8 times ____/
11111111 00000001 xxxxxxxx .. xxxxxxxx = 0..0FFFFFFFFFFFFFFFFFh
      \added_/ \_____ 9 times ____/
...
```

# Simple Block-Tree Structure

Data are organized into blocks, which are sequence of values and binary blobs.

```
UBENatural AttributePartLength (Begin with A)
A UBENatural DataPartLength (Begin with D)
A UBNumber Attribute1
A ...
A UBNumber AttributeN
D Data (Binary blob)
```

If there are attributes, data part is processed as a sequence of blocks. Otherwise it is data blob. Document is single block and following data are processed as extended blob.

# Hierarchical Types

First two attributes are interpreted as a block type. First means group index and second block index which allows using similar approach as XML-Namespaces.

Meaning of indexes is declared using built-in basic blocks. Single format can be declared from MIB-like catalog or using internal or external source (for example URL).

Also attributes have declared types.

# Specification's Catalog

As a basic source for specification MIB-like catalog is available. Specifications are organized into tree with following optional extensions:

Name – Tag name for human-readable visualizations

Description – Textual description of meaning

Icon – Graphical icon for given type

Comment – More detailed description of block meaning and more...

# Transformations

Protocol will support data transformations and transformation scripts in future.

Meaning of data will be declared through transformations.

Interface for automatic transformations will be available as well as a service providing operating system integration.

# Multilevel Architecture

Protocol is split into multiple isolated levels with rising complexity. This means that you don't have to support all protocol functions.

Level 0: Numeric encoding and block-tree structure

Level 1: Block types and basic blocks

Level 2: Attribute types

Level 3: Transformations

more in later releases...

# Support

Support for following issues will be mentioned:

- Parsing Support
- Streaming Support
- Basic Tools
- Catalog
- Services

# Parsing Support

Four parser types should be implemented:

- Event Parser – Sending single tokens, sender driven
- Pull Parser – Getting single tokens, receiver driven
- Object Model Parser – Building complete copy in memory with object bindings
- Command Parser – Combination of all previous with support for modification postponement

# Streaming Support

To support protocol streaming, following issues shall be supported:

- Input/Output Streams – Support for input and output streams
- Serialization – Support for conversion object values into streams
- Operators, filters and buffers – Interfaces for data processing

# Basic Tools

For testing purposes following applications shall be created:

- XB Editor – General editor for files encoded using XBUP Protocol
- XB Shell – Shell-like browser of XBUP files content
- XB Service – Service providing local copy of catalog
- XB Manager – Tool for accessing XBUP Service and catalog

# Catalog

Catalog service should be provided as well.

General service should be available on internet providing specifications and allowing developers to store their own format declarations.

Local services should download relevant parts of catalog on demand.

# Services

Highly customizable modular system should allow parallel usage of transformation modules with customizable preferences and upgrade services.

Service should also provide optional interface for remote invoking via network with support for computer clusters.

# Conclusion

Project is still in its early phase and lot of work have to be done. So far it seems possible to accomplish given goals.

If you are interested, visit project web page:

<http://xbup.sf.net/>

Any contributions are welcomed.