libpackedobjects tutorial

libpackedobjects - an XML schema-aware compression library

Table of Contents

1 Introduction 1

1.1 What is libpackedobjects? 1

1.2 Key features 1

1.3 Use case 1

1.4 Limitations 1

2 Installation 2

2.1 Installing libpackedobjects 2

2.2 Further reading 2

3 Getting started 3

3.1 Quick start 3

3.2 Writing a schema 3

4 Data types 4

4.1 Simple types 4

4.2 Complex types 4

Index 5

1 Introduction

1.1 What is libpackedobjects?

libpackedobjects is a C library which can be used to efficiently compress an XML DOM by using the information provided by a corresponding XML Schema. The tool is designed for writing network protocols which strive to minimise the amount of data communicated. This is similar to something like Protocol Buffers but using XML. As well as encode efficiently the tool validates all data according to the schema.

libpackedobjects is based on libxml2 and therefore should run on any system that libxml2 runs on.

1.2 Key features

- Very efficient encoding size yet fast
- Good choice of data types including the ability to apply range and size constraints
- Fully dynamic including the ability to change the protocol at runtime
- Simple API with two main function calls
- Highly portable: supported on embedded Linux and mobile platforms

1.3 Use case

libpackedobjects is being incorporated into an XML messaging framework for specific use on wireless networks.

1.4 Limitations

libpackedobjects is not a general purpose document compression tool. It is intended to be used in an application that generates XML that you wish to communicate over a network. As such it provides a simple DOM-based API for encoding and decoding structured data. The compression technique used is based on applying knowledge of the data types specified in a schema to provide better performance over statistical compression techniques. Therefore, you must write a valid schema for your data. However, the schema serves the purpose of formalising the network protocol and provides validation. Thus we think having a schema is a good thing!

2 Installation

2.1 Installing libpackedobjects

To install from the latest source:

```
git clone git://gitorious.org/libpackedobjects/libpackedobjects.git
cd libpackedobjects
autoreconf -i
./configure
make
make check
sudo make install
```

2.2 Further reading

3 Getting started

3.1 Quick start

3.2 Writing a schema

4 Data types

4.1 Simple types

4.2 Complex types

Index

C

Complex types 4

F

Further reading 2

I

Installing libpackedobjects 2

K

Key features 1

L

Limitations 1

Q

Quick start 3

S

Simple types 4

U

Use case 1

W

What is libpackedobjects 1
Writing a schema 3