XF 1.0 (Data Exchange Format)
Формат обмена данными
Ревизия 6 (14 апреля 2008 г.)
http://xfhome.org

Волков Максим, Севастополь 2006-2007 (mailto:max@xfhome.org)

1. Введение

XF (data eXchange Format) – универсальный, легкий и переносимый формат представления данных в текстовом виде, который просто воспринимается человеком и обрабатывается программами.
XF переносим и не зависит от программного и аппаратного обеспечения. Документы XF могут быть прочитаны программой на любом языке программирования, поддерживающем эту технологию.
В отличие от XML, XF не предназначен для разметки текстовых документов. XF – это универсальный формат для хранения данных с произвольной структурой.

2. Основные свойства XF

- XF – это универсальный формат обмена данных, способный представлять данные любой структуры. Возможно создавать форматы обмена данными и протоколы, основанные на XF (использующие XF для представления данных).
- Документы XF описывают древовидную структуру данных, состоящую из элементов, которые могут имеют имя, тип (класс) и значение. Модель данных не зависит от их представления, но XF предлагает форму их представления – формат XF.
- XF легко читаем для человека, его просто создавать и редактировать вручную. Документы XF представляют собой обычный текст.
- XF предельно компактен.
- XF прост в обработке программами.
- XF переносим и не зависит от аппаратной и программной платформы. Он может поддерживаться программами на любом языке программирования.
- XF полностью поддерживает спецификацию Unicode для представления данных на различных языках.

3. Основные понятия

Документ XF описывает структуру данных, представляющую собой дерево, состоящее из элементов, каждый из которых имеет имя, класс (тип) и значение.
Элементы идентифицируются по имени и местоположению в дереве. Среди прямых

потомков элемента имена элементов уникальны. Порядок элементов не важен, и потомки каждого элемента образуют множество (не список).
Дерево XF не имеет единого корня: глобальных элементов (не являющихся потомками) может быть неограниченное число.

4. Запись документов XF

Документ XF может быть представлен в 7-битной кодировке ASCII, в 8-битной кодировке Unicode UTF-8, или в 16-битной кодировке Unicode UTF-16 (с прямым или обратным порядком байт). Документ может состоять из букв, цифр, специальных символов, разделённых проблемами, знаками табуляции и перевода строки.

5. Комментарии

Комментарии – это фрагменты текста, игнорируемые программами, работающими с XF и предназначенными для прочтения человеком.

В XF есть два типа комментариев – однострочные и многострочные.

1. Однострочные комментарии начинаются с символа * (звёздочка), и продолжаются до конца строки.
2. Многострочные комментарии могут занимать неограниченное число строк. Они заключаются между символами /* и */.

Примеры однострочных комментариев

```
* Это однострочный комментарий
xf version = "1.0"; * это тоже однострочный комментарий
```

Примеры многострочных комментариев

```
/* Это многострочный комментарий */

/* и это тоже
   многострочный комментарий */
```

6. Строки

Имена элементов, их значения и классы представлены в виде строк символов Unicode. Строки могут содержать любые символы кроме нулевого (код 0) и иметь неограниченную длину. Строка не может быть пуста (должна содержать хотя бы один символ).

Существует два вида строк:

1. Простая строка. Записывается непосредственно и может состоять только из латинских строчных и прописных букв, цифр, дефиса и знака подчёркивания: (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z - _).

Примеры простых строк

test, element1

123

_

some_element

class

2. Расширенная строка. Заключается с двух сторон в двойные кавычки (#0022 в Unicode-представлении). Внутри могут содержаться ЛЮБЫЕ символы, кроме двойных кавычек, которые используются как признак начала/конца строки. В строках этого типа можно использовать различные комбинации символов со специальным назначением (escape-последовательности):

\\	знак \ (обратный слеш)
*	двойная кавычка (")
\n	перевод строки
\t	табуляция
\s	пробел
\x;	Unicode-символ с шестнадцатеричным кодом «x». Шестнадцатеричный код может состоять из одной или более шестнадцатеричных цифр (0 1 2 3 4 5 6 7 8 9 a b c d e f A B C D E F). Возможное количество цифр не ограничено.

Надо отметить, что в подобных строках могут содержаться пробелы, переводы строки и любые другие символы (кроме двойной кавычки).

Примеры расширенных строк

"some string"

"!@#$%^&*()"

"one line\nanother line"

"эта строка
содержит перевод строки"

Внимание! Строки в XF не могут быть пустыми. Любая строка в XF должна содержать хотя бы один символ (запись "" не является корректной строкой).

7. Запись элементов

Как уже отмечалось, любой элемент имеет имя, может принадлежать некоторому классу и содержать значение. Имя элемента, класс и значение могут быть записаны в виде строки (простой, расширенной или составной).

Описание элемент может быть записано в следующих формах:

имя
имя = значение
имя : класс
имя : класс = значение
имя = значение : класс

Примеры записи элементов:

```
height: int = 200

size = 4

name = "Ritchie Blackmore"

note: string

comment = "this is a comment" : any_class
```

В случае если значение или класс элемента определены в документе неоднократно, программа-обработчик XF может выдать предупреждение, но документ считается корректно сформированным. В этом случае классом элемента считается тот класс, с которым он описан последним, значением элемента считается то значение, с которым он описан последним.

8. Запись дерева

Дерево XF подразумевает, что каждый элемент либо принадлежит некоторому другому элементу, либо не принадлежит ни одному из элементов (такой элемент называется глобальным).
Документ XF состоит из набора предложений. Каждое предложение состоит из описания одного или нескольких элементов. Каждый следующий элемент в предложении принадлежит предыдущему. Предложения отделяются друг от друга точкой с запятой (;). Количество подряд стоящих точек с запятой не ограничено.

Примеры предложений

```
a b c d;
font color:string = Black;
price:int = 5 currency = "EUR";
```

В примерах:

- ***элемент "d" принадлежит элементу "c", принадлежащему элементу "b". Элемент "b" принадлежит элементу "a";***
- ***элемент "color" (описано его значение) принадлежит элементу "font";***
- ***элемент "currency" принадлежит элементу "price" (описаны классы и значения элементов).***

Элементы, принадлежащие одному и тому же элементу, можно заключать в фигурные скобки { и }. Набор предложений, заключённых в фигурные скобки, называется блок. Конец блока также является концом предложения, и после него не обязательно ставить точки с запятой (;). Перед закрывающей фигурной скобкой в блоке также не обязательно ставить точки с запятой.

Пример использование блоков

```
Songs: Playlist
 {
  song1
   {
    title = "Transylvania";
    artist = "Iron Maiden";
    album = "Iron Maiden";
    year = 1980;
   }
  song2
   {
    title = "Fade to Black";
    artist = "Metallica";
    album = "Ride The Lightning";
    year = 1884
   };
 }
```

В приведённом примере элемент "Songs" содержит элементы "song1" и "song2", каждый из которых включает элементы "title", "artist", "album" и "year", содержащие значения. Как видно из примера, в конце блоков и перед закрывающей фигурной скобкой точки с запятой не обязательны.

Пример комбинирования блоков и обычной записи предложений

```
user login = "Floyd_Rose";
user name = "Max Volkov";
point coordinates X = "1.5";
point coordinates Y = "2.0";

program version = "1.0" {
  major = "1";
  minor = "0";
 }

test { { { a } } };
```

В приведённом примере обе записи user представляют собой один и тот же элемент, имеющий потомков "login" и "name". Элементы "major" и "minor" принадлежат элементу "version". Из описания элемента "test" видно, что можно открывать несколько блоков вложенных друг в друга.

9. Атрибуты документов (рекомендация)

В документах XF может присутствовать элемент, который хранит служебную информацию о версии формата XF и типе документа. Этот элемент – «xf», расположенный в корне документов. Включать элемент «xf» в документы не обязательно, но рекомендуется. Рекомендуется, чтобы элемент «xf» был записан в самом начале документа.

Элемент «xf» может содержать несколько потомков, значения которых определяют характеристики документа:

- **version** – номер версии формата XF (текущая версия – "1.0"); рекомендуется включать во все документы для совместимости версий.
- **class** – тип документа или программы (символьное имя); рекомендуется включать во все документы идентификации типа документа.
- **comment** – комментарий. Может содержать комментарий о документе, предназначенный для прочтения человеком.

Пример атрибутов документа:

```
xf version =  "1.0";
xf class = "Document.Example";

xf comment = "Комментарий!!!";
```

10. Разработчики

1. **Волков Максим** (Max Volkov, max@xfhome.org). Идея, разработка модели данных и формата XF, документации, ANSI C-библиотеки для работы с XF (XFLib), официального сайта проекта (http://xfhome.org).

Приложение. Формальное описание грамматики XF

В данной спецификации грамматика XF описывается с помощью расширенной формы Бакуса-Науру. Каждое правило грамматики определяет один символ в форме

```
символ ::= выражение
```

В выражении в правой части правила для сопоставления строк из одного и более символов используются следующие выражения:

#xN
где N - шестнадцатеричное целое число, выражение соответствует символу набора ISO/IEC 10646, каноническое (UCS-4) кодовое значение которого, в случае интерпретации в виде беззнакового двоичного числа, имеет указанное значение;

[a-zA-Z], [#xN-#xN]
соответствует любому символу со значением в указанном диапазоне (включительно);

[^a-z], [^#xN-#xN]
соответствует любому символу со значением вне указанного диапазона;

[^abc], [^#xN#xN#xN]
соответствует любому символу со значением, не входящим в число указанных символов;

'строка'
соответствует литеральной строке, соответствующей строке, данной в одинарных кавычках;

(выражение)
выражение обрабатывается как единица и может объединяться, как описано в данном списке;

A B
соответствует A, за которым следует B;

A | B
соответствует A или B, но не обоим вместе;

A - B
соответствует любой строке, соответствующей A, но не соответствующей B;

A+
соответствует одному или нескольким экземплярам A;

A*
соответствует любому числу экземпляров A;

[описание]
текстовое описание символа

Грамматика XF

EOF ::= *[*конец файла *(*документа ***XF)]***

Character ::= #x0000000A | #x0000000D | #x00000009 | ([#x00000020-#xFFFFFFFF] - (#x0000FFFF | #x0000FFFE)

Space ::= #x00000020
Tab ::= #x00000009
EOL ::= #x0000000A | #x0000000D | #x0000000D #x0000000A | #x0000000A #x0000000D

StringCharacter ::= Character - ('"' | '\')
HexadecimalDigit ::= [0123456789abcdefABCDEF]

ShortComment ::= '*' (Character - EOL)* (EOL | EOF)
LongComment ::= '/*' LongCommentText '*/'
LongCommentText ::= *[* Некоторый текст, не содержащий комбинацию двух символов *'*/'* *]*

S ::= (EOL | Space | Tab | LongComment | ShortComment)*

LongString ::= '"' (Character)+ '"'

ShortString::=
[abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_-]+

NodeName ::= ShortString | LongString
NodeValue ::= ShortString | LongString
NodeClass ::= ShortString | LongString

Node ::= S NodeName S |
S NodeName S ':' S NodeClass S |
S NodeName S '=' S NodeValue S |
S NodeName S ':' S NodeClass S '=' NodeValue S |
S NodeName S '=' S NodeValue S ':' NodeClass S

Sentenses ::= (S Node S | S ';' S)+
Block ::= '{' (S Sentenses S | S Block S | S)* '}'
Document ::= (S Block S | S Sentenses S | S)*