

Спецификация формата данных KAML

Версия 1.8 31.03.2019

Назначение

Формат KAML (Key Array Markup Language, язык разметки индексированных массивов) является промежуточным форматом представления (данных), которые удобно представлять преимущественно в виде таблиц и массивов. К таким данным можно отнести исходные данные для программ расчетов с использованием метода конечных элементов, полевым методом и другими подобными методами математической физики.

В качестве представления данных KAML в качестве его основы могут использоваться разные форматы представления - простой текстовый формат (plain text), JSON, XML, реляционные базы данных и т.д.

При использовании для данных KAML текстового формата, данные небольших по объему прикладных задач могут создаваться в текстовых редакторах и восприниматься подготовленным пользователем без необходимости использования программ для визуализации, что удобно при изучении структуры данных и отладке программ и скриптов при их разработке.

KAML является форматом представления абстракций данных и не содержит каких либо классов и типов для описания прикладных задач.

KAML разработан для компактного и эффективно обрабатываемого представления структур объектов с насыщенным геометрическим представлением и простой системой параметров. Примерами использования KAML являются описание объектов BIM-моделей, расчетных моделей строительных объектов и т.п.

KAML является форматом описания абстрактных структур данных на основе других языков описания, таких как TXT, JSON, JSON5, XML и т.п.

Структура KAML подобна структуре таблиц в реляционных базах данных, поэтому предполагается простая конвертация данных KAML в реляционные базы данных и их обработка SQL и подобными средствами

Основными «опорными» форматами для KAML являются «плоский текст» TXT и JSON

Файлы на KAML могут преобразовываться в двоичный формат по какой-либо спецификации (например UBJSON), и упаковываться (архивироваться)

На основе KAML разрабатываются прикладные форматы (протоколы) - описания каких-либо прикладных данных, структура которых подходит для эффективного применения KAML.

Лицензирование

Формат KAML разработан ООО "Ситис" в 2018 году для обмена информацией между расчетными программными комплексами.

Формат является открытым и свободным для любого использования любыми заинтересованными лицами

Спецификация формата

Ниже приведено краткое описание структур данных и правил их описания.

Для более полного понимания структуры данных рекомендуется ознакомиться с какими-либо примерами моделей в форматах, использующих KAML для представления прикладных данных.

Массивы

Массивы данных могут быть следующих типов индексации:

1-«ключевой» массив. Индексация выполняется по ключу, заданному в записях. Ключ в записях всегда задан. (по умолчанию)

2-«порядковый» массив. Индексация выполняется по порядковому номеру записи. Ключи в записях отсутствуют. Ключом является порядковый номер записи.

3-«порядко-ключевой» массив. Внутренняя индексация при записи в JSON выполняется по порядковому номеру записи. Внешняя индексация выполняется по ключу. Ключ в записях всегда задан. Для текстового KAML этот тип соответствует типу 1.

Структура данных

1. Все данные в KAML представляются в виде массива объектов, называемых записями.
2. Типом массива является назначение данных для прикладной задачи, данные которой сохраняются в файле KAML.
3. Тип массива всегда имеет однозначно заданную структуру для соответствующих прикладных данных.
4. В файле KAML может быть использован только один экземпляр массива для каждого типа массива.
5. Записи в массиве индексируются по индексу, состоящему из одного целочисленного ключа, заданного в записи, или по порядковому номеру записи.
6. Нумерация записей для индексирования осуществляется по порядку без пропусков начиная с 1.
7. Записи состоят из следующих объектов
 - 7.1. Ключ записи – целое положительное число, уникальное для массива. То есть в массиве не может быть двух записей с одинаковым значением ключа. Для порядковых массивов ключ отсутствует.
 - 7.2. Список параметров записи. Параметром записи может быть только целое или действительное число. Список может быть произвольной длины. Список может отсутствовать.
 - 7.3. Список свойств записи. Список пар «ИмяСвойства=ЗначениеСвойства», заключенный в фигурные скобки. Список свойств может отсутствовать. Для свойства в зависимости от их использования в прикладных данных могут быть задано несколько значений, т.е. несколько пар «имя=значение». Имя прикладных (пользовательских) свойств не следует начинать или заканчивать знаком подчеркивания, чтобы не путать с служебными свойствами и макропеременными.
 - 7.4. Название. Произвольный текст в кавычках. Может отсутствовать.
 - 7.5. Описание. Произвольный текст после знака «//». Может отсутствовать.
 - 7.6. В текстовом написании после списка параметров допускается произвольная последовательность задания названия и списка свойств, то есть в начале может указываться и название, и список свойств.
8. Для массива кроме списка его записей могут задаваться следующие объекты:
 - 8.1. Тип индексации массива
 - 8.2. Атрибут – целое положительное число. Может рассматриваться как ключ для массива, состоящего из атрибутированных массивов соответствующего типа.

- 8.3. Список свойств. Список пар «ИмяСвойства=ЗначениеСвойства», заключенный в фигурные скобки. Список свойств может отсутствовать. Для свойства в зависимости от их использования в прикладных данных могут быть задано несколько значений, т.е. несколько пар «имя=значение».
- 8.4. Свойства массива подразделяются на служебные и прикладные. Служебные свойства массива описывают методы хранения и интерпретации данных в массиве. Прикладные свойства используются для описания прикладных характеристик хранящихся данных. Имена служебных свойств начинаются с знака «_». Имя прикладных (пользовательских) свойств не следует начинать или заканчивать знаком подчеркивания, чтобы не путать с служебными свойствами и макропеременными.
- 8.5. Название. Произвольный текст в кавычках. Может отсутствовать.
- 8.6. Описание. Произвольный текст после знака «//». Может отсутствовать.
- 8.7. После списка параметров допускается произвольная последовательность задания названия и списка свойств, то есть в начале может указываться и название, и список свойств.
- 9. Разделителями элементов списков параметров являются пробел и знак табуляции
- 10. Разделителями элементов списков свойств массива и записи являются пробел, знак табуляции, знаки перевода и возврата строки
- 11. В тексте могут быть заданы комментарии. Комментарием считается часть строки после знака «///».

Шаблон описания массива в KAML TXT

Разделитель – пробел, табуляция и т.п.

Вся информация о записи – в одной строке

Ключевой массив (строка начинается с «@»):

```
@ТИПМАССИВА атрибут «названиеМассива» {списокСвойствМассива} // описание /// комментарий
  ключЗаписи СписокПараметровЗаписи «названиеЗаписи» {списокСвойствЗаписи} // описание ///
  комментарий
&
```

Порядковый массив (строка начинается с «\$»):

```
$ТИПМАССИВА атрибут «название» {списокСвойствМассива} // описание /// комментарий
  СписокПараметровзаписи «название» {списокСвойствЗаписи} // описание /// комментарий
&
```

Допускается также другая последовательность задания названий и списков свойств:

```
@ТИПМАССИВА атрибут {списокСвойствМассива} «названиеМассива» // описание /// комментарий
  ключЗаписи СписокПараметровЗаписи {списокСвойствЗаписи} «названиеЗаписи» // описание ///
  комментарий
&
```

ТИПМАССИВА является обозначением, которое может быть номером или строкой названия.

ТипыМассива задаются при описании структур прикладных данных с помощью формата KAML

НомерТипаМассива - целое положительное число, как правило трехзначное. Служебные массивы имеют двухзначные номера

НазваниеТипаМассива - произвольная строка, соответствующая номеруТипаМассива. , не являющаяся строкой какого-либо числа. Как правило название массива задается в виде краткой строки из заглавных латинских символов.

То есть обозначения @100 и @ARRAYNAME равнозначны для массива с номером 100 и именем ARRAYNAME.

Имена массивов могут быть переопределены, номера массивов задаются в структуре описания прикладных данных жестко и не переопределяются.

Основным в идентификации массива является номер массива. Название массива может быть переопределено на различных языках.

Соответствие названий типов массивов и номеров массивов может быть указано и/или преопределено в служебном массиве с номером 20 - массиве \$ATNAME_

При разработке скриптов и программ следует как правило использовать номера массивов, полученные с использованием словаря \$ATNAME_

Шаблон описания массива в KAML JSON

Мнемоника:

Имена из двойные буквы – объекты, относящиеся к массиву

Имена с символом @ - объекты, относящиеся к списку записей массива, с индексацией по ключу – для ключевых и порядко-ключевых массивов

Имена с символом \$ - объекты, относящиеся к списку записей массива, с индексацией по номеру записи – для порядковых и порядко-ключевых массивов

Имена с символом % - объекты, относящиеся к служебным данным для ускорения обработки, как правило описываю взаимное соответствие ключей и порядковых номеров. (опционально, могут отсутствовать)

Если атрибут массива не задан, то имя соответствующего объекта задается без «##»

Шаблон для ключевого массива

```
“МАССИВ##атрибут”: { “ IndexType_”:1,  
  “NN”: название массива  
  “KK”: описание массива  
  “ТТ”:[ИмяСвойстваМассива, значение]] – служебные свойства массива  
  “РР”:[ИмяСвойстваМассива, значение]] – пользовательские свойства массива  
  “А$”: [  
    [списокПараметров]  
  ],  
  “Р@”: [  
    [ключ, [ИмяСвойстваЗаписи, значение]]  
  ],  
  “N@”: [  
    [ключ, название]  
  ],  
  “D@”: [  
    [ключ, описание]  
  ],  
  “K%”: [  
    [ключ, значение]  ]
```

```
        [ ключ, порядковый номер записи в A@] /// в этом списке ключи отсортированы по
возрастанию значений
    ],
}
}
```

Шаблон для порядкового массива

```
“МАССИВ##атрибут”: { “ IndexType_”:2,
    “NN”: название массива
    “KK”: описание массива
    “ТТ”:[ИмяСвойстваМассива, значение] – служебные свойства массива
    “РР”:[ИмяСвойстваМассива, значение] – пользовательские свойства массива
    “A$”: [
        [списокПараметров]
    ],
    “P$”: [
        [порядковый номер в A$, [ИмяСвойстваЗаписи, значение]
    ],
    “N$”: [
        [порядковый номер A$, название]
    ],
    “D$”: [
        [ порядковый номер A$, описание]
    ],
}
}
```

Шаблон для порядко-ключевого массива

```
“МАССИВ##атрибут”: { “IndexType_”:3,
    “NN”: название массива
    “KK”: описание массива
    “ТТ”:[[ПараметрТипа, значение]] – служебные параметры
    “РР”:[[параметрМассива,значение]] – пользовательские параметры
    “A@”: [
        [ключ, список чисел]
    ],
    “P$”: [
        [порядковый номер, [ИмяСвойстваЗаписи, значениеСвойства]
    ]
    ],
    “N$”: [
        [порядковый номер, название]
    ],
    “D$”: [
        [ порядковый номер, описание]
    ],
    “K%”: [
        [ ключ, порядковый номер записи в A@] /// в этом списке ключи отсортированы по
возрастанию значений
    ],
}
```

}

Тип индексации массива (ключевой, порядковый, порядко-ключевой) описывается элементом (объектом) "IndexType_".

Служебные свойства

Если в таблице служебных свойств свойство не задано, то принимается значение свойства по умолчанию.

Группа свойств «Округление и преобразование»

Данные свойства используются для возможности использования целочисленной арифметики и хранения данных для ускорения обработки, сокращения размера файлов и улучшения восприятия текста человеком.

MC_ (Multiple Coefficient) коэффициент преобразования, на который умножаются числа в списке параметров в записях объекта A@ массива, при распаривании массива. По умолчанию 1.0.

RP_ (Rounding to Power) – порядок округления – целое число, соответствующее степени порядка округления чисел, записанного в списке параметров записей массива. Например: 0- округление до целых, 2 – округление до сотен (10 в 2 степени), -2 – округление до процентов, то есть 10^{-2} степени . округление производится при сохранении данных в формате KAML. По умолчанию -999 – округление не проводится.

PD_ (PropertyNameDictionary) – указывает, что для данного массива вместо имен свойств записей указаны их коды по словарю имен. Данная возможность описана в разделе «Специальные массивы. Словари»

Пример (JSON): [{"MC_": 0.01}, {"RP_": 0}] – результатом будут целые числа, соответствующие сотым частям чисел в списках массива. При распаривании прочитанное число должно быть умножено на 0.01. Данная запись соответствует округлению чисел до 0.01.

Пример (JSON): [{"MC_": 1}, {"RP_": -2}] – результатом будут числа, соответствующие числам в списках массива, записанных с округленным до $0.01=10^{-2}$. При распаривании прочитанное число должно быть умножено на 1. Данная запись соответствует округлению чисел до 0.01. То есть до процентов. Число 1 в списке чисел массива KAML будет соответствовать значению 1.00

Служебные массивы

Описания служебных массивов

Язык разметки описания прикладных данных, в том числе служебных массивов, приведен в приложении А.

Контрольные данные

Версии форматов KAML и описаний прикладных данных

\$VER_ (030) := Версия ревизия списокОпций “НазваниеФормата”

НазваниеФормата – название формата. В том числе используется для подключения словарей параметров и имен типов массивов соответствующего формата.

списокОпций – список специальных опций языка, использованных в описании данных. Поскольку разные библиотеки обработки KAML могут поддерживать не все опции, то они могут контролировать совместимость данных до начала их разбора

Опции реализации препроцессора KAML:

200 - макропеременные

210 - работа с константами

220 - работа с константами и выражениями

230 – работа с списками и диапазонами

240 – работа с параметрическими пространствами

300-диалоги

310-подстановки

320-запросы

400-словари

410 – работа со словарями

Пример:

```
$VER_  
1 4 202 “KAML” //202-используется препроцессор с поддержкой арифметических  
выражений  
1 0 101 102 “SitisSolaris” // Ситис:Солярис. 101 - данные для расчета инсоляции, 102-KEO.  
&
```

Пример:

```
$VER_  
1 4 “KAML”  
1 2 “BIMDump”  
1 0 102 “SitisBlockDump”  
&
```

Обработчики

Таблица для информации о программах и обработчиках, создавших или изменивших данные в файле

\$HANDLER_ (035) := >время Версия ревизия списокПараметров “НазваниеОбработчика”

Системное время

\$TIMESTAMP_ (040) := год месяц дата час минута секунда

Препроцессорная обработка

Перед разбором файла по спецификации KAML может быть выполнена предварительная обработка файла с использованием информации из специальных служебных массивов. Служебные массивы имеют номера типов массивов от 1 до 99, имена типов зачисляются знаком «_».

Обработчик для предварительной обработки называется препроцессором KAML, предварительная обработка – препроцессорной.

Функции препроцессора:

Валидация – проверка соответствия значений макропеременных и других настроек значениям свойств и параметров в массивах файла.

Преобразование – присвоение свойствам и параметрам массивов значений, соответствующих значений макропеременных и словарных кодов, заданных в файле

Приведение – разбор ключей вызова программы препроцессора, присвоение заданных значений макропеременным, преобразований файла.

Запрос- вывод диалоговой формы для ввода пользователем значений макропеременных, приведение файла

Препроцессор использует служебные файлы (файлы словарей, кодировок и т.п.), находящиеся в указанной папке

Словари (перспектива, пока не реализовываем)

Для удобства написания текстов с описанием данных в KAML предусмотрена возможность при препроцессорной обработке файла заменять строковые константы численными значениями из массива \$DICT_, который называется словарем мнемоники, или просто словарем. Как правило словарь разрабатывается при описании структур прикладных данных на KAML и указывается в соответствующем описании

Строки таблицы словаря параметров

\$PARDICT_ (010) := ТипЗамены ТипМассива Позиция значение предыдущееЗначение предыдущаяПозиция «Название»

ТипЗамены:: 1-предыдущееЗначение не учитывается; 2-предыдущаяПозиция, то есть позиция предыдущего параметра, не учитывается. То есть предыдущая позиция равна Позиции, уменьшенной на 1; 3-учитываются все поля

Пример:

1 100 3 0 0 “КОД” /// если третий параметр записи в массиве с номером типа 100 равен строке “КОД”, то этот параметр будет задан равным 100

2 100 3 5 0 “КОД” /// если третий параметр записи в массиве с номером типа 100 равен строке “КОД”, то этот параметр будет задан равным 100 при условии, что второй параметр равен 5

3 100 3 5 1 “КОД” /// если третий параметр записи в массиве с номером типа 100 равен строке “КОД”, то этот параметр будет задан равным 100 при условии, что первый параметр равен 5

При описании структур данных на KAML можно использовать следующие записи заданных в словаре кодов для каких-либо параметров массивов:

КОД (значениеПараметра), или значениеПараметра(КОД)

Форма записи выбирается по усмотрению разработчика, в зависимости от его привычек или удобства изложения.

Если в файлах KAML предполагается хранить много записей с большим количеством свойств с одинаковыми длинными именами, то возможно уменьшать размер файла за счет замены имени свойства его порядковым номером в словаре имен свойств.

Кодировка имен свойств записей массива задается служебным параметром PD_.

Массив словаря имеет вид:

\$PROP_ (015) := «ИмяСвойства»

Пример:

@A

```
1 { myPrettyPropertyName=1023}
3 { myPrettyPropertyName=1023}
9 { myPrettyPropertyName=1023 ВтороеКрасивоеПонятноеНазваниеСвойства=0}
1122 { myPrettyPropertyName=1023}
```

@B

```
1 { ПонятноеИмяСвойства=да}
3 { ПонятноеИмяСвойства =нет}
```

\$PROP_

“myPrettyPropertyName”
“ВтороеКрасивоеПонятноеНазваниеСвойства”
“ПонятноеИмяСвойства”

@A {PD_=1}

```
1 { 1=1023}
3 { 1=1023}
9 { 1=1023 2=0}
1122 { 1=1023}
```

@B {PD_=1}

```
1 { 3=да}
3 { 3=нет}
```

В служебном массиве **\$ATNAME_** можно определить, переопределить или указать на другом языке названия типов массивов.

Также этот массив может использоваться для извлечения соответствия номеров типов массивов и названий типов массивов для обработки результатов построцессорами

\$ATNAME_ (020) := НомерТипаМассива “НазваниеТипаМассива”

Переменные (перспектива, пока не реализовываем)

В KAML можно задавать переменные, присваивать им арифметические или строковые значения. При препроцессорной обработке переменные, заданные в списках параметров, значениях свойств, названиях и описаниях, заменяются строку, соответствующую значению переменной. При приведении значений переменной к строковому виду учитываются параметры округления и преобразования, заданные служебными свойствами соответствующего массива. Поскольку осуществляется замена текстовой строки названия на текстовую строку значения, такая замена является макроподстановкой, и переменные называются макропеременными.

Макропеременная в тексте должна быть отдельным словом, отделенным от других слов пробелами, знаком табуляции или перевода строки, служебными символами «={}», кавычками.

@VAR_ (040) := “имяМакропеременной” {...}

Свойства:

c – число или строка, присваиваемая переменной

e - арифметическое или строковое выражение, присваиваемое переменной значения, написанное по правилам языка JavaScript

k-признак, что значение данной переменной может присваиваться через ключи вызова программы препроцессора

u-код диалоговой формы, в которой реализуется редактирование данных пользователем. Может быть задано несколько таких свойств для разных диалоговых форм

d-значение по умолчанию

ИмяМакропеременная всегда начинается с «_»

Например:

```
@VAR_
```

```
10 "_A" {c=100}
```

```
20 "_M" {e=_A*2}
```

```
30 "_Num" {c=10}
```

```
@ABC "Вариант номер _Num"
```

```
1 2 3 4 5 6 7 {_A.param=2} /// это ключевой массив. Параметры начинаются с второй позиции
```

```
$BCD // Описание к варианту _Num
```

```
1 2 3 4 5 6 7 {_A.param=2} /// это порядковый массив. Параметры начинаются с первой позиции
```

```
$CDE
```

```
1 2 3 { a=0 _A.prop=a}
```

```
@DEF
```

```
1 2 3 {_A.key=1}
```

Будет соответствовать массивам

```
@ABC "Вариант номер 10" {_Num.name.pos=3}
```

```
1 2 100 4 5 6 7 {_A.param=2} /// это ключевой массив. Параметры начинаются с второй позиции
```

```
$BCD {_Num.dscr.pos=4} // Описание к варианту 10
```

```
1 100 3 4 5 6 7 {_A.param=2} /// это порядковый массив. Параметры начинаются с первой позиции
```

```
$CDE
```

```
1 2 3 { a=100 _A.prop=a}
```

```
@DEF
```

```
100 2 3 {_A.key=1}
```

При препроцессорной обработке в тексте KAML можно определять значения макропеременных выражением «макропеременная=арифметическое или строковое выражение» и вставлять макропеременные вместо параметров, в названиях и описаниях. Макропеременная в тексте должна быть отдельным словом, отделенным от других слов пробелами

Такая опция реализации препроцессора имеет номер 230

Пример:

_M=100
_C= 100 +_M/2

@ABC
1 2 _M 4 5 6 7 "Запись номер _M" // Описание к записи _M

\$BCD
1 _C 3 4 5 6 7

\$CDE
1 2 3 { a=_M}

Массив диалогов ввода и редактирования значений переменных

\$DVFORM_(045) := кодДиалога >списокПеременных

ПРИЛОЖЕНИЕ А

KAML-S - язык разметки для описания схем прикладных данных в формате KAML

Структуры прикладных данных с использованием KAML описываются как список массивов.

Для краткости описания структура массива и структура записей массива записываются в одной строке.

Для этого используются специальные служебные литералы - ">", ":", не являющиеся элементами формата KAML, и использующиеся только для описания структуры данных на KAML

Информация о параметрах "шапки" массива от информации о параметрах записей массива отделяется знаком «:=».

До знака «:=» приводится описание заголовка ("шапки") массива - тип, номер, название массива, атрибуты массива

После знака «:=» приводится описание параметров записи массива

Первый символ строки описания массива обозначает тип массива (@-ключевой, \$-порядковый), после него указывается имя типа массива.

После имени типа массива в круглых скобках указывается ключ типа массива.

После типа массива может быть указано имя атрибута массива. Как правило это ссылка на записи в каком-либо массиве.

Знак «>» в начале имен и названий обозначает (ссылка на запись в массиве с указанным именем». Если массив ключевой, то в этом поле указывается значение ключа. Если массив порядковый – то указывается порядковый номер записи.

В структуре прикладных данных не указываются явно названия и описания массивов и их записей. Они могут задаваться всегда.

Пример:

@Проект(100)

///Данная запись обозначает что в ключевом массиве номер 100 с названием Проект нет записей. Для массива может быть указано имя, описание и свойства.

\$Перо (102) := R G B A типПера

///Данная запись обозначает что в порядковом массиве номер 102 с названием Перо записи состоят из пяти числовых параметров - R, G, B, A, .

///Для массива может быть указано имя, описание и свойства.

///Для записей также может быть указано имя, описание и свойства.

@Точка (103) := X Y

///Данная запись обозначает что в ключевом массиве номер 103 с названием Точка. Записи состоят номера ключа записи и из двух числовых параметров - X, Y .

///Для массива может быть указано имя, описание и свойства.

///Для записей также может быть указано имя, описание и свойства.

@Линия (104) := >списокТочек

///Данная запись обозначает что в порядковом массиве номер 104 с названием Линия. Записи состоят из номера ключа и списка чисел, содержащих ссылки на номера (ключи) точек.

///Для массива может быть указано имя, описание и свойства.

///Для записей также может быть указано имя, описание и свойства.

@Фигура (105) := >Перо >списокЛиний

///Данная запись обозначает что в порядковом массиве номер 103 с названием Точка записи состоят из двух числовых параметров - X, Y .

///Для массива может быть указано имя, описание и свойства.

///Для записей также может быть указано имя, описание и свойства.

/// dx dy dz – смещение координат оболочки и каркаса для данного объекта. То есть можно задать оболочку и каркас типового этажа, потом его использовать для представления нескольких этажей. По умолчанию dx dy dz равны 0

\$Группа (200) НомерГруппы := >списокФигур

///Данная запись обозначает что в ключевом массиве номер 200 с названием Здание и атрибутом НомерГруппы, записи состоят из списка чисел, который представляет список номеров фигур.

///Для массива может быть указано имя, описание и свойства.

///Для записей также может быть указано имя, описание и свойства.