

## objString — альтернативное представление строки как объекта

Иногда требуется собрать строку, в частности если собирать вручную JSON или XML. Если для получения результата придётся произвести порядка 100 тысяч операций сложения строк, то становятся заметны потери времени на проведение этих операций средствами RSL. В качестве альтернативы при оптимизации одного интеграционного решения была создана эта библиотека.

### Класс TObjString

Класс предназначен для хранения собираемой строки вместо переменной типа String.

#### Конструктор TObjString([str1:String[, str2:String,..]])

Конструктору может быть передано любое количество аргументов, в результате объект будет проинициализирован конкатенацией всех аргументов с типом строка. Если не передано ни одного аргумента, то объект инициализируется пустой строкой.

```
var s1=TObjString(); // ""
var s2=TObjString("Hello, world"); // "Hello, world"
var s3=TObjString("Hello, ", "world"); // "Hello, world"
```

#### Метод Add(str1:String[, str2:String,...])

Добавляет в конец строки объекта все аргументы, имеющие тип строки.

```
var s=TObjString(); // ""
s.Add("Hello, ", "world"); // "Hello, world"
```

#### Метод AddJsonKeyValue(key:String, value:Variant)

Метод рассчитан на то что собирается строка JSON. Добавляется пара ключ и значение. При повторном вызове сперва добавляется разделитель запятая. Ключ **key** должен быть строкой, значение **value** может быть примитивным типом, при этом он преобразуется к соответствующему типу JSON.

```
var s=TObjString(); // ""
s.AddJsonKeyValue("Name", "Ромашкин Александр");
s.AddJsonKeyValue("BirthDate", Date(20,11,1998));
s.AddJsonKeyValue("Salary", $123456.78);
s.AddJsonKeyValue("пдл", true);
// "Name": "Ромашкин Александр", "BirthDate": "1998-11-20", "Salary": 123456.78, "пдл": true
```

#### Метод ResetJsonKeyValue()

Влияет на метод AddJsonKeyValue. После вызова ResetJsonKeyValue первый вызов AddJsonKeyValue не будет сопровождаться установкой разделителя.

#### Метод InsertAfter(str:String)

Добавляет строку **str** в конец строки объекта.

#### Метод InsertBefore(str:String)

Добавляет строку **str** в начало строки объекта.

#### Метод Clear()

Очищает строку объекта.

#### Свойство Value:String

Только чтение. Возвращает строку объекта.

### Свойство String:String

Только чтение. Возвращает строку объекта, псевдоним Value.

### Свойство Length:Integer

Только чтение. Возвращает длину строки объекта.

### Свойство IsEmpty:Bool

Только чтение. Если строка объекта пуста, то true. Иначе false. Пустой считается только строка нулевой длины, например, если строка состоит из пробела, то вернётся false.

### Метод Save(fileName:String):Bool

Записывает строку в файл. Путь к файлу передаётся в качестве аргумента. В случае успеха возвращается true, иначе false.

### Свойство DateFormat:String

Чтение запись. Возвращает или устанавливает формат даты. Влияет на метод AddJsonKeyValue. Определяет в каком формате значение типа дата будет помещено в строку. Спецификация формата приведена в описании функции [strftime](#). По умолчанию установлен формат ГГГГ-ММ-ДД (%Y-%m-%d).

### Метод SetDateFormatDMY()

Устанавливает свойство DateFormat в значение ДД.ММ.ГГГГ (%d.%m.%Y).

### Функция objstrClaibrateLocaleA(a:String)

Влияет на метод AddJsonKeyValue. Должна быть вызвана один раз перед использованием метода, в качестве аргумента передаётся заглавная русская буква А. Определяет, в какой кодировке в DLM передаются строки. Такие сложности необходимы, только если макрос планируется использовать из-под RSLSvc.

## О скорости

Замеры на тестовом стенде добавления подстроки в строку 1, 10 и 100 тысяч раз.

	1 тыс.	10 тыс.	100 тыс.
RSL	0:00:00.03	0:00:01.66	0:03:14.44
DLM	0:00:00.00	0:00:00.01	0:00:00.06

Таким образом, если предполагается до тысячи конкатенаций, то использование DLM не имеет смысла, до 10 тысяч — только если существуют жесткие требования по времени ответа, на больших количествах применение DLM даёт ощутимое преимущество.