

**ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»**

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

**«МОЙОФИС КОМПЛЕКТ СРЕДСТВ РАЗРАБОТКИ (SDK)»**

**ИНФОРМАЦИЯ О НОВОЙ ВЕРСИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
(RELEASE NOTES)**

**3.4**

**Версия 1**

**На 13 листах**

**Дата публикации: 03.06.2025**

**Москва**

**2025**

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам.

Товарные знаки «МойОфис» и «MyOffice» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

Ни при каких обстоятельствах нельзя истолковывать любое содержимое настоящего документа как прямое или косвенное предоставление лицензии или права на использование товарных знаков, логотипов или знаков обслуживания, приведенных в нем. Любое несанкционированное использование этих товарных знаков, логотипов или знаков обслуживания без письменного разрешения их правообладателя строго запрещено.

## СОДЕРЖАНИЕ

1 Надстройки .....	4
1.1 Работа со стилями .....	4
1.2 Работа со сносками .....	4
1.3 Работа с таблицами и ячейками .....	4
1.4 Работа с изображениями .....	5
1.5 Другие изменения .....	5
2 Макрокоманды Lua .....	6
2.1 Работа со стилями .....	6
2.2 Работа со сносками .....	6
2.3 Работа с таблицами и ячейками .....	6
2.4 Работа с изображениями .....	7
2.5 Другие изменения .....	7
3 Document API C++ .....	8
3.1 Работа со стилями .....	8
3.2 Работа со сносками .....	8
3.3 Работа с таблицами и ячейками .....	8
3.4 Работа с изображениями .....	9
3.5 Другие изменения .....	9
4 Document API C# .....	10
4.1 Работа со стилями .....	10
4.2 Работа со сносками .....	10
4.3 Работа с таблицами и ячейками .....	10
4.4 Работа с изображениями .....	11
4.5 Другие изменения .....	11
5 Document API Python .....	12
5.1 Работа со стилями .....	12
5.2 Работа со сносками .....	12
5.3 Работа с таблицами и ячейками .....	12
5.4 Работа с изображениями .....	13
5.5 Другие изменения .....	13

## 1 НАДСТРОЙКИ

### 1.1 Работа со стилями

Добавлена новая таблица `TextStyle`, которая представляет собой стиль абзаца в текстовом документе. Методы этой таблицы позволяют задать и получить название стиля, настройки форматирования абзаца (`ParagraphProperties`) и текста (`TextProperties`), а также стиль следующего абзаца. Дополнительно таблица `TextStyle` содержит метод `TextStyle:createChild` для создания дочернего стиля и метод `TextStyle:getParent` для получения родительского стиля.

Для описания коллекции стилей добавлена таблица `TextStyles`. Стили в этой коллекции определяются по имени или идентификатору `PredefinedTextStyle` (для стандартных стилей). Метод `TextStyles:create` позволяет добавить новый стиль в коллекцию. Для получения стиля используется метод `TextStyles:get`.

Добавлен метод `Document:getTextStyles` для получения стилей текстового документа.

Задать стиль абзаца можно с помощью метода `Paragraph:setStyle`. Чтобы получить стиль абзаца, используйте методы `Paragraph:getStyle` и `Paragraph:getResolvedStyle`.

### 1.2 Работа со сносками

Теперь вы можете взаимодействовать со сносками и концевыми сносками в текстовом документе: получить коллекцию сносок, вставить сноску и концевую сноску, получить позицию сноски в тексте и диапазон содержимого сноски. Новые таблицы `FootnoteEndnote` и `FootnotesEndnotes` представляют собой сноску и их коллекцию.

Методы `Position:insertFootnote` и `Position:insertEndnote` позволяют вставить сноску и концевую сноску, соответственно.

Чтобы получить все сноски диапазона, используется метод `Range:getFootnotesEndnotes`.

### 1.3 Работа с таблицами и ячейками

- Добавлен метод `CellRange:intersect`, который возвращает область пересечения двух диапазонов ячеек.
- Теперь вы можете определить, содержит ли ячейка данные или настройки форматирования, с помощью методов `Cell:isContentEmpty` или `Cell:isEmpty`.

Аналогичные методы добавлены для диапазона ячеек: `CellRange:isContentEmpty` и `CellRange:isEmpty`.

- Новый метод `Cell:getResolvedBorders` позволяет получить настройки границ ячейки с учетом границ окружающих ячеек.

## **1.4 Работа с изображениями**

- Теперь вы можете вставлять изображения на лист табличного документа с помощью метода `Table:insertImage`.
- Добавлен метод `Image:replaceURL`, который позволяет заменить изображение в документе.

## **1.5 Другие изменения**

- Добавлен метод `NamedExpression:setName`, который позволяет переименовать именованный диапазон.
- Новый статический метод `Position:getBefore` возвращает позицию в текстовом документе, которая находится перед заданной таблицей.

## 2 МАКРОКОМАНДЫ LUA

### 2.1 Работа со стилями

Добавлена новая таблица `TextStyle`, которая представляет собой стиль абзаца в текстовом документе. Методы этой таблицы позволяют задать и получить название стиля, настройки форматирования абзаца (`ParagraphProperties`) и текста (`TextProperties`), а также стиль следующего абзаца. Дополнительно таблица `TextStyle` содержит метод `TextStyle:createChild` для создания дочернего стиля и метод `TextStyle:getParent` для получения родительского стиля.

Для описания коллекции стилей добавлена таблица `TextStyles`. Стили в этой коллекции определяются по имени или идентификатору `PredefinedTextStyle` (для стандартных стилей). Метод `TextStyles:create` позволяет добавить новый стиль в коллекцию. Для получения стиля используется метод `TextStyles:get`.

Добавлен метод `Document:getTextStyles` для получения стилей текстового документа.

Задать стиль абзаца можно с помощью метода `Paragraph:setStyle`. Чтобы получить стиль абзаца, используйте методы `Paragraph:getStyle` и `Paragraph:getResolvedStyle`.

### 2.2 Работа со сносками

Теперь вы можете взаимодействовать со сносками и концевыми сносками в текстовом документе: получить коллекцию сносок, вставить сноску и концевую сноску, получить позицию сноски в тексте и диапазон содержимого сноски. Новые таблицы `FootnoteEndnote` и `FootnotesEndnotes` представляют собой сноску и их коллекцию.

Методы `Position:insertFootnote` и `Position:insertEndnote` позволяют вставить сноску и концевую сноску, соответственно.

Чтобы получить все сноски диапазона, используется метод `Range:getFootnotesEndnotes`.

### 2.3 Работа с таблицами и ячейками

- Добавлен метод `CellRange:intersect`, который возвращает область пересечения двух диапазонов ячеек.
- Теперь вы можете определить, содержит ли ячейка данные или настройки форматирования, с помощью методов `Cell:isContentEmpty` или `Cell:isEmpty`.

Аналогичные методы добавлены для диапазона ячеек: `CellRange:isContentEmpty` и `CellRange:isEmpty`.

- Новый метод `Cell:getResolvedBorders` позволяет получить настройки границ ячейки с учетом границ окружающих ячеек.

## 2.4 Работа с изображениями

- Теперь вы можете вставлять изображения на лист табличного документа с помощью метода `Table:insertImage`.
- Добавлен метод `Image:replaceURL`, который позволяет заменить изображение в документе.

## 2.5 Другие изменения

- Добавлен метод `NamedExpression:setName`, который позволяет переименовать именованный диапазон.
- Новый статический метод `Position:getBefore` возвращает позицию в текстовом документе, которая находится перед заданной таблицей.

## 3 DOCUMENT API C++

### 3.1 Работа со стилями

Добавлен новый класс `TextStyle`, который представляет собой стиль абзаца в текстовом документе. Методы этого класса позволяют задать и получить название стиля, настройки форматирования абзаца (`ParagraphProperties`) и текста (`TextProperties`), а также стиль следующего абзаца. Дополнительно класс `TextStyle` содержит метод `TextStyle::createChild` для создания дочернего стиля и метод `TextStyle::getParent` для получения родительского стиля.

Для описания коллекции стилей добавлен класс `TextStyles`. Стили в этой коллекции определяются по имени или идентификатору `PredefinedTextStyle` (для стандартных стилей). Метод `TextStyles::create` позволяет добавить новый стиль в коллекцию. Для получения стиля используется метод `TextStyles::get`.

Добавлен метод `Document::getTextStyles` для получения стилей текстового документа.

Задать стиль абзаца можно с помощью метода `Paragraph::setStyle`. Чтобы получить стиль абзаца, используйте методы `Paragraph::getStyle` и `Paragraph::getResolvedStyle`.

### 3.2 Работа со сносками

Теперь вы можете взаимодействовать со сносками и концевыми сносками в текстовом документе: получить коллекцию сносок, вставить сноску и концевую сноску, получить позицию сноски в тексте и диапазон содержимого сноски. Новые классы `FootnoteEndnote` и `FootnotesEndnotes` представляют собой сноску и их коллекцию.

Методы `Position::insertFootnote` и `Position::insertEndnote` позволяют вставить сноску и концевую сноску, соответственно.

Чтобы получить все сноски диапазона, используется метод `Range::getFootnotesEndnotes`.

### 3.3 Работа с таблицами и ячейками

– Добавлен метод `CellRange::intersect`, который возвращает область пересечения двух диапазонов ячеек.



- Теперь вы можете определить, содержит ли ячейка данные или настройки форматирования, с помощью методов `Cell::isEmpty` или `Cell::isContentEmpty`. Аналогичные методы добавлены для диапазона ячеек: `CellRange::isEmpty` и `CellRange::isContentEmpty`.
- Новый метод `Cell::getResolvedBorders` позволяет получить настройки границ ячейки с учетом границ окружающих ячеек.

## 3.4 Работа с изображениями

- Теперь вы можете вставлять изображения на лист табличного документа с помощью метода `Table::insertImage`.
- Добавлен метод `Image::replaceURL`, который позволяет заменить изображение в документе.

## 3.5 Другие изменения

- Добавлен метод `NamedExpression::setName`, который позволяет переименовать именованный диапазон.
- Новый статический метод `Position::getBefore` возвращает позицию в текстовом документе, которая находится перед заданной таблицей.

## 4 DOCUMENT API C#

### 4.1 Работа со стилями

Добавлен новый класс `TextStyle`, который представляет собой стиль абзаца в текстовом документе. Методы этого класса позволяют задать и получить название стиля, настройки форматирования абзаца (`ParagraphProperties`) и текста (`TextProperties`), а также стиль следующего абзаца. Дополнительно класс `TextStyle` содержит метод `TextStyle.createChild` для создания дочернего стиля и метод `TextStyle.getParent` для получения родительского стиля.

Для описания коллекции стилей добавлен класс `TextStyles`. Стили в этой коллекции определяются по имени или идентификатору `PredefinedTextStyle` (для стандартных стилей). Метод `TextStyles.create` позволяет добавить новый стиль в коллекцию. Для получения стиля используется метод `TextStyles.get`.

Добавлен метод `Document.getTextStyles` для получения стилей текстового документа.

Задать стиль абзаца можно с помощью метода `Paragraph.setStyle`. Чтобы получить стиль абзаца, используйте методы `Paragraph.getStyle` и `Paragraph.getResolvedStyle`.

### 4.2 Работа со сносками

Теперь вы можете взаимодействовать со сносками и концевыми сносками в текстовом документе: получить коллекцию сносок, вставить сноску и концевую сноску, получить позицию сноски в тексте и диапазон содержимого сноски. Новые классы `FootnoteEndnote` и `FootnotesEndnotes` представляют собой сноску и их коллекцию.

Методы `Position.insertFootnote` и `Position.insertEndnote` позволяют вставить сноску и концевую сноску, соответственно.

Чтобы получить все сноски диапазона, используется метод `Range.getFootnotesEndnotes`.

### 4.3 Работа с таблицами и ячейками

- Добавлен метод `CellRange.intersect`, который возвращает область пересечения двух диапазонов ячеек.
- Теперь вы можете определить, содержит ли ячейка данные или настройки форматирования, с помощью методов `Cell.isContentEmpty` или `Cell.isEmpty`.

Аналогичные методы добавлены для диапазона ячеек: `CellRange.isEmpty` и `CellRange.isContentEmpty`.

- Новый метод `Cell.getResolvedBorders` позволяет получить настройки границ ячейки с учетом границ окружающих ячеек.

## 4.4 Работа с изображениями

- Теперь вы можете вставлять изображения на лист табличного документа с помощью метода `Table.insertImage`.
- Добавлен метод `Image.replaceURL`, который позволяет заменить изображение в документе.

## 4.5 Другие изменения

- Добавлен метод `NamedExpression.setName`, который позволяет переименовать именованный диапазон.
- Новый статический метод `Position.getBefore` возвращает позицию в текстовом документе, которая находится перед заданной таблицей.
- Теперь вы можете использовать в качестве текстовых параметров строки, символы которых кодируются двумя байтами (например, кириллицу), без дополнительных преобразований.

## 5 DOCUMENT API PYTHON

### 5.1 Работа со стилями

Добавлен новый класс `TextStyle`, который представляет собой стиль абзаца в текстовом документе. Методы этого класса позволяют задать и получить название стиля, настройки форматирования абзаца (`ParagraphProperties`) и текста (`TextProperties`), а также стиль следующего абзаца. Дополнительно класс `TextStyle` содержит метод `TextStyle.createChild` для создания дочернего стиля и метод `TextStyle.getParent` для получения родительского стиля.

Для описания коллекции стилей добавлен класс `TextStyles`. Стили в этой коллекции определяются по имени или идентификатору `PredefinedTextStyle` (для стандартных стилей). Метод `TextStyles.create` позволяет добавить новый стиль в коллекцию. Для получения стиля используется метод `TextStyles.get`.

Добавлен метод `Document.getTextStyles` для получения стилей текстового документа.

Задать стиль абзаца можно с помощью метода `Paragraph.setStyle`. Чтобы получить стиль абзаца, используйте методы `Paragraph.getStyle` и `Paragraph.getResolvedStyle`.

### 5.2 Работа со сносками

Теперь вы можете взаимодействовать со сносками и концевыми сносками в текстовом документе: получить коллекцию сносок, вставить сноску и концевую сноску, получить позицию сноски в тексте и диапазон содержимого сноски. Новые классы `FootnoteEndnote` и `FootnotesEndnotes` представляют собой сноску и их коллекцию.

Методы `Position.insertFootnote` и `Position.insertEndnote` позволяют вставить сноску и концевую сноску, соответственно.

Чтобы получить все сноски диапазона, используется метод `Range.getFootnotesEndnotes`.

### 5.3 Работа с таблицами и ячейками

- Добавлен метод `CellRange.intersect`, который возвращает область пересечения двух диапазонов ячеек.
- Теперь вы можете определить, содержит ли ячейка данные или настройки форматирования, с помощью методов `Cell.isContentEmpty` или `Cell.isEmpty`.

Аналогичные методы добавлены для диапазона ячеек: `CellRange.isEmpty` и `CellRange.isContentEmpty`.

- Новый метод `Cell.getResolvedBorders` позволяет получить настройки границ ячейки с учетом границ окружающих ячеек.

## 5.4 Работа с изображениями

- Теперь вы можете вставлять изображения на лист табличного документа с помощью метода `Table.insertImage`.
- Добавлен метод `Image.replaceURL`, который позволяет заменить изображение в документе.

## 5.5 Другие изменения

- Добавлен метод `NamedExpression.setName`, который позволяет переименовать именованный диапазон.
- Новый статический метод `Position.getBefore` возвращает позицию в текстовом документе, которая находится перед заданной таблицей.