

unit uofficedll.pas v.0,7 Beta 13/12/2011

Динамическая библиотека для автоматизации создания отчётов в M\$ Word XP/2003 в Delphi/Lazarus. Для использования необходимо в проект подключить модуль uworddll.pas (в Lazarus также необходимо подключить модуль Variants и модуль ComObj).

Описание функций для работы с M\$ Word

function CentimetersToPoints(cm: Real): Real;

Переводит сантиметры в точки

procedure NewDocument(var Wrd: Variant; visible: Boolean);

Создает новый документ. Параметр visible указывает, будет ли отображаться окно Word'a на экране

procedure PageMargins(l, r, t, b: Single; var wrd: Variant);

Устанавливает поля страницы

procedure PageOrientation(orientation: Integer; var wrd: Variant);

Устанавливает альбомную/книжную ориентацию страницы. wdOrientPortrait = 0; //книжна
wdOrientLandscape = 1; //альбомна

procedure HFDistance(h, f: Single; var wrd: Variant);

Устанавливает отступы для верхнего и нижнего колонтитулов

Procedure PageSize(w, h: Single; var wrd: Variant);

Устанавливает размер страницы в сантиметрах. Для A4 w=29.7, h=21

procedure SetOnPage(style: Integer; var Wrd: Variant);

Устанавливает количество и размещение листов на странице. wdNormalPage = 0; //1 к 1 - стандартно
wdTwoOnOne = 1; //2 стор. на 1 wdBookFold = 2; //брошюра

procedure PageAlign(align: Integer; var wrd: Variant);

Устанавливает вертикальное выравнивание текста на странице. wdAlignVerticalTop = 0; //по верхнему краю
wdAlignVerticalCenter = 1; //по центру wdAlignVerticalJustify = 2; //разтянуть по высоте
wdAlignVerticalBottom = 3; //по нижнему краю

procedure NewPage(wrd: Variant);

Вставить разрыв страницы в документ

procedure FontName(name: ShortString; var wrd: Variant);

Устанавливает название шрифта (напр. 'Courier New')

procedure FontSize(sz: Integer; var Wrd: Variant);

Устанавливает размер для текущего шрифта

procedure FontBold(Bold: Boolean; var Wrd: Variant);

Устанавливает жирность для текущего шрифта

procedure FontItalic(Italic: Boolean; var Wrd: Variant);

Устанавливает курсив для текущего шрифта

procedure FontUnderlined(underlined: Boolean; var wrd: Variant);

Устанавливает подчеркивание для текущего шрифта

procedure FontShadowed(Shadowed: Boolean; var Wrd: Variant);

Устанавливает тень для текущего шрифта

procedure FontColor(Color: Integer; var Wrd: Variant);

Устанавливает цвет для текущего шрифта wdAuto = 0; wdBlack = 1; wdBlue = 2; wdTurquoise = 3; wdBrightGreen = 4; wdPink = 5; wdRed = 6; wdYellow = 7; wdWhite = 8; wdDarkBlue = 9; wdTeal = 10; wdGreen = 11; wdViolet = 12; wdDarkRed = 13; wdDarkYellow = 14; wdGray50 = 15; wdGray25 = 16; wdByAuthor = -1; wdNoHighlight = 0;

procedure FontSuperScript(super: Boolean; var Wrd: Variant);

Устанавливает верхние индексы

procedure FontSubScript(sub: Boolean; var Wrd: Variant);

Устанавливает нижние индексы

procedure FontSpacing(spacing: Single; var Wrd: Variant);

Устанавливает межсимвольный интервал: 1, 1.5 и т.д.

procedure FontScaling(scaling: Integer; var Wrd: Variant);

Устанавливает масштаб для текущего шрифта в %

procedure FontPosition(position: Single; var Wrd: Variant);

Устанавливает смещение текста вверх (положительные значения) или вниз (отрицательные значения) в пт

procedure AddText(s: ShortString; var wrd: Variant);

Вставить строку

procedure AddParagraph(var wrd: Variant);

Начать новый абзац

procedure ParagraphAlign(align: Integer; var Wrd: Variant);

Установить выравнивание абзаца по ширине wdAlignParagraphLeft = 0; wdAlignParagraphCenter = 1; wdAlignParagraphRight = 2; wdAlignParagraphJustify = 3;

procedure ParagraphLineSpace(space: Integer; var wrd: Variant);

Установить междустрочный интервал: wdLineSpaceSingle = 0; wdLineSpace1pt5 = 1; wdLineSpaceDouble = 2; wdLineSpaceAtLeast = 3; wdLineSpaceExactly = 4; wdLineSpaceMultiple = 5;

procedure ParagraphIndents(l, r: Single; var wrd: Variant);

Установить отступы абзаца слева и справа (в см)

procedure ParagraphSpaces(top, bottom: Single; var wrd: Variant);

Установить отступы абзаца сверху и снизу

Procedure ParagraphFirstLine(indent: Single; var wrd: Variant);

Установить отступ первой строки абзаца (в см)

procedure AddTabPosition(pos: Single; var wrd: Variant);

Вставить позицию табуляции в pos см

procedure DefaultTabPos(pos: Single; var wrd: Variant);

Установить позицию табуляции по умолчанию в pos см

procedure ClearAllTabs(var wrd: Variant);

Очистить все позиции табуляции

```
procedure CreateTable( Col, Row: Integer; var wrd: Variant );
```

Создать таблицу с атрибутами по умолчанию

```
procedure CreateTableEx( Col,Row : Integer; DefaultTableBehavior,  
                        AutoFitBehavior : Integer; var wrd : Variant );
```

Создать таблицу с расширенными параметрами. Допустимые значения:

DefaultTableBehavior:	AutoFitBehavior:
wdWord8TableBehavior = 0;	wdAutoFitFixed = 0;
wdWord9TableBehavior = 1;	wdAutoFitContent = 1;
	wdAutoFitWindow = 2;

```
Procedure SetColWidth( wid: Single; var wrd: Variant );
```

Установить ширину текущего столбца в таблице. Использовать после вызова [AddText\(\)](#), иначе значение игнорируется

```
Procedure SetRowHeight( h: Single; var wrd: Variant );
```

Установить высоту текущей строки в таблице.

```
procedure GotoRight( cells: Integer; var wrd: Variant );
```

Перейти на cells ячеек вправо. Если таблица закончилась, вставляет новую строку

```
procedure GotoLeft( cells: Integer; var wrd: Variant );
```

Перейти на cells ячеек влево.

```
procedure GotoUp( lines: Integer; var wrd: Variant );
```

Перейти на lines строк вверх

```
procedure GotoDown( lines: Integer; var wrd: Variant );
```

Перейти на lines строк вниз.

```
Procedure MergeCellsR( count: Integer; var Wrd: Variant );
```

Объединить указанные ячейки вправо. Курсор должен находиться в первой из них

```
Procedure MergeCellsD( count: Integer; var Wrd: Variant );
```

Объединить указанные ячейки вниз. Курсор должен находиться в первой из них

```
procedure DeleteRow( var Wrd: Variant );
```

Удалить текущую строку таблицы

```
procedure DeleteCol( var Wrd: Variant );
```

Удалить текущий столбец таблицы

```
Procedure ExitTable( wrd: Variant );
```

Выйти из таблицы

```
procedure CellTextOrientation( orient: Integer; wrd: Variant );
```

Направление текста в таблице: wdTextOrientationHorizontal = 0; wdTextOrientationUpward = 2;
wdTextOrientationDownward = 3; wdTextOrientationVerticalFarEast = 1;
wdTextOrientationHorizontalRotatedFarEast = 4;

```
procedure InsertHeader( hdr: ShortString; var wrd: Variant );
```

Вставить верхний колонтитул

```
procedure InsertFooter( ftr: ShortString; var wrd: Variant );
```

Вставить нижний колонтитул

```
procedure SetWordVisible( var wrd: Variant );
```

сделать Ворд видимым на экране

```
function CheckWordVersion( var wrd: Variant ): Boolean;
```

проверить может ли использоваться установленная версия Ворда

```
function SaveDocAs( file_: Shortstring; var wrd : Variant): Boolean;
```

сохранить созданный документ с указанным именем и путем

```
function CloseDoc( var wrd : Variant ): Boolean;
```

закрыть документ

```
function CloseWord( var wrd : Variant ): Boolean;
```

выход из Word'a

```
function PrintDialogWord( var wrd : Variant ): Boolean;
```

вызов диалога печати

Описание функций для работы с M\$ Excell

```
procedure NewXlsDocument( var xls : Variant; visible : Boolean);
```

Создает новый документ Excel

```
procedure OpenXlsDocument( var xls : Variant; xlsfile : ShortString);
```

Открыть указанный документ Excel

```
function GetXlsWorkBook( var xls : Variant; idx : Integer): Variant;
```

Получить ссылку на активную книгу

```
function GetXlsWorkBookSheet( var WorkBook : Variant;  
                               idx : Integer) :Variant;
```

Получить ссылку на активный лист

```
procedure SetCellValue( var Xls : Variant;  
                        CellName : ShortString; value : ShortString);
```

Записать значение как текст в указанную ячейку

```
procedure SetCellValueInteger( var Xls : Variant;  
                               CellName : ShortString; value : Integer);
```

Записать значение как целое число в указанную ячейку

```
procedure SetCellValueFloat( var Xls : Variant;  
                             CellName : ShortString; value : Double);
```

Записать значение как число с плавающей точкой в указанную ячейку

```
procedure SetCellValueDate( var Xls : Variant;  
                            CellName : ShortString; value : TDateTime);
```

Записать значение как дату в указанную ячейку