

# WDRReportCom

## User Manual

**Version 1.5**

2009-08

*Copyright© 2007-2009 LJZsoft Corporation*

All rights reserved

# Contents

<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
1.1 OVERVIEW.....	1
1.2 FEATURES.....	1
<b>CHAPTER 2 INSTALLATION AND STARTUP .....</b>	<b>3</b>
2.1 SOFTWARE REQUIREMENTS.....	3
2.2 INSTALLING WDRPTCOM .....	3
2.3 UNINSTALLING WDRPTCOM .....	4
2.4 WDRPTCOM.DLL.....	4
2.5 WORDREPORT.EXE.....	5
2.6 RUN-TIME FILES.....	6
<b>CHAPTER 3 QUICK START.....</b>	<b>7</b>
3.1 LEARNING HOW TO USE WDRPTCOM.....	7
3.2 SAMPLE DATABASE .....	7
3.3 SAMPLES.....	8
3.4 CREATING A REPORT PROGRAMMATICALLY.....	9
3.5 CREATING A REPORT WITH WORDREPORT.EXE .....	10
<b>CHAPTER 4 REPORT TEMPLATES .....</b>	<b>12</b>
4.1 ABOUT REPORTS .....	12
4.2 ABOUT REPORT TEMPLATES.....	12
4.3 WORD BASIC CONCEPTS.....	12
4.3.1 Documents.....	13
4.3.2 Headers and Footers.....	13
4.3.3 Tables.....	13
4.3.4 Bookmarks.....	13

4.3.5 Page Breaks .....	13
4.3.6 Graphics and Diagrams.....	14
4.3.7 Inline Pictures and Floating Pictures .....	14
4.3.8 Charts.....	14
4.3.9 Formatting .....	15
4.3.10 Fields .....	16
4.4 TABLE REPORTS .....	17
4.4.1 About Table Reports.....	17
4.4.2 Creating a Table for a Fixed Table Report.....	17
4.4.3 Creating a Table for a Variable Table Report.....	18
4.4.4 Formatting Cells.....	19
4.4.5 Irregular Tables .....	20
4.4.6 Referencing Cells.....	21
4.4.7 Referencing Tables.....	22
4.4.8 Formatting Cells for Pictures .....	23
4.5 FORM REPORTS.....	24
4.5.1 About Form Reports.....	24
4.5.2 Creating Merge Fields or Quote Fields.....	25
4.5.3 Defining Ranges.....	26
4.5.4 Formatting Fields for Pictures.....	27
4.6 CHARTS .....	28
4.6.1 About Charts.....	28
4.6.2 Creating a Blank Chart using Microsoft Graph .....	29
4.6.3 Creating a Blank Chart using Microsoft Excel.....	29
4.6.4 Referencing Charts .....	31
<b>CHAPTER 5 API REFERENCE .....</b>	<b>33</b>
5.1 OBJECTS .....	33
5.1.1 WDRReport Object.....	33

5.2 METHODS .....	33
5.2.1 <i>FixTableReport Method</i> .....	33
5.2.2 <i>VarTableReport Method</i> .....	36
5.2.3 <i>GroupTableReport Method</i> .....	39
5.2.4 <i>FormReport Method</i> .....	43
5.2.5 <i>MSGraphChart Method</i> .....	47
5.2.6 <i>ExcelChart Method</i> .....	49
5.2.7 <i>SetDocVariable Method</i> .....	52
5.2.8 <i>WordReport Method</i> .....	54
5.2.9 <i>GetTableByIndex Method</i> .....	55
5.2.10 <i>GetTableByBookmark Method</i> .....	56
5.2.11 <i>GetChartByIndex Method</i> .....	56
5.2.12 <i>GetChartByBookmark Method</i> .....	57
5.3 EVENTS .....	58
5.3.1 <i>BeforeConnect Event</i> .....	58
5.3.2 <i>TemplateOpen Event</i> .....	59
5.3.3 <i>ReportComplete Event</i> .....	60
5.3.4 <i>FunctionBeforeExectue Event</i> .....	60
5.3.5 <i>FunctionAfterExectue Event</i> .....	61
5.3.6 <i>FunctionProgress Event</i> .....	62
5.4 ERROR MESSAGES .....	63
<b>CHAPTER 6 WRF FILES.....</b>	<b>65</b>
6.1 USING WRF FILES .....	65
6.1.1 <i>About WRF files</i> .....	65
6.1.2 <i>Using a WRF file with WDRreport Object</i> .....	65
6.1.3 <i>Using a WRF file in command line</i> .....	66
6.1.4 <i>Creating a WRF file</i> .....	66
6.1.5 <i>Using parameters</i> .....	67

6.1.6 Converting files.....	70
6.2 WRF FILE REFERENCE.....	72
6.2.1 WRF File Format.....	72
6.2.2 [Data Source] Section.....	74
6.2.3 [FILE] Section .....	76
6.2.4 [PARAMETER] Section .....	77
6.3 FUNCTION REFERENCE .....	78
6.3.1 Fixed Table Report.....	78
6.3.2 Variable Table Report.....	81
6.3.3 Group Table Report .....	84
6.3.4 Form Report.....	88
6.3.5 MSGraph Chart .....	91
6.3.6 Excel Chart .....	93
6.3.7 DocVariable.....	95
6.3.8 ExecSQL.....	97
<b>CHAPTER 7 ADVANCED REPORTS .....</b>	<b>99</b>
7.1 EXECUTING MULTIPLE SQL STATEMENTS .....	99
7.2 USING FORMULA FIELDS .....	102
7.3 SORTING, GROUPING AND TOTALING .....	104
7.3.1 Sorting data.....	104
7.3.2 Totaling .....	105
7.3.3 Grouping data and Subreports.....	105
7.3.4 Subtotaling .....	106
7.4 PICTURES.....	107
7.4.1 Inserting pictures into a report template.....	107
7.4.2 Inserting pictures into a report .....	107
<b>CHAPTER 8 HINTS AND TIPS.....</b>	<b>109</b>

<b>CHAPTER 9 FORMAT EXPRESSIONS IN DATA CELLS .....</b>	<b>115</b>
A.1 FORMATS FOR NUMERIC VALUES .....	115
A.2 FORMATS FOR STRING VALUES .....	119
A.3 FORMATS FOR DATE/TIME VALUES .....	120
<b>CHAPTER 10 LICENSE AND SUPPORT .....</b>	<b>123</b>
10.1 LICENSE .....	123
10.2 TECHNICAL SUPPORT .....	124

# Chapter 1 Introduction

## 1.1 Overview

WDRReportCom is a solution that generates reports using Microsoft Word. Using Microsoft Word and WDRReportCom, you can create all kinds of reports quickly and easily. WDRReportCom includes an ActiveX DLL and an executable file that can be used to develop your applications. It will significantly accelerate your application development.

WDRReportCom is a template-based solution. To create a report, you need to create a report template file first. The report template file is a Microsoft Word document that defines the layouts and formats of a report. WDRReportCom retrieves data from data source and fills data into Word documents.

## 1.2 Features

WDRReportCom includes the following features:

- Using Microsoft Word as your reporting tool

Just use Microsoft Word as your reporting tool. You design reports like layouts, formats and styles directly using Microsoft Word. And you will get reports in Microsoft Word spreadsheet format as a result. Microsoft Word is powerful, flexible and familiar. You do not need to buy and learn extra reporting tools.

- Making report template directly using Microsoft Word

The main advantage of using WDRReportCom is based on the fact that all formatting is done directly in Microsoft Word. You can take full advantage of Microsoft Word including text formatting, tables, fields, pictures and graphics, drawing, page setup, headers and footers, preview and printing, VBA, macros,

and more.

- Accessing to databases using SQL

WDRReportCom executes SQL statements to extract data from database.

Supports all type SQL: DML, DDL and DCL. Multiple SQL statements can be executed in one report building process. You can perform queries on databases, insert data into databases, and create database objects like tables.

The power of SQL can be harnessed for maximum efficiency in reporting.

- Using ADO to access and manipulate data sources

Using ADO, WDRReportCom can access and manipulate a wide variety of data sources such as Oracle, DB2, Sybase, Informix, Microsoft SQL Server, Teradata, MySQL, Microsoft Access, dBase.

- Integrating Microsoft Word into your application

WDRReportCom includes an ActiveX DLL for building application. Developers can save time and meet their users' needs by integrating the report processing power of WDRReportCom into their applications.

- Command line program

WDRReportCom includes a command line program WordReport.exe. You can use the program to create reports too. It does not require programming. It is enough if you know how to use Microsoft Word and how to write SQL.

- Various reporting capabilities

WDRReportCom provides various reporting capabilities including sorting data, grouping data, subreports, totaling and summarizing data, formatting, charting and pictures. It is easy to create simple reports, and, you can create complex reports.

## **Chapter 2 Installation and Startup**

### **2.1 Software Requirements**

Microsoft Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows 2003, Windows Vista or later.

Microsoft Office 97/98, Office 2000, Office XP, Office 2003 or later.

### **2.2 Installing WDRReportCom**

Run the installation program, and follow the instructions to complete WDRReportCom installation. For Windows Vista, the data folder should be different from the application folder.

If you don't have Microsoft Office installed, please install it first.

If your environment is Windows 95/98 and Office 97, and you don't have VB6.0 run-time files installed, please install it. For Windows 2000, Windows XP, Windows 2003 and Office 2000 or later, you do not need to install VB6.0 run-time files because they are included in OS and Office. To install VB6.0 run-time files, just run vbrun60sp5.exe, and follow the instructions.

If you don't have ODBC Driver installed for the database you want to access, please install it.

If your OS is Windows 95/98 and you don't have Microsoft Data Access Components 2.0 (MDAC\_TYP) or later installed, please install it. For Windows 2000, Windows XP and Windows 2003, you do not need to install MDAC\_TYP because it is preinstalled in OS. To install MDAC\_TYP, just run mdac\_typ.exe, and follow the instructions.

## 2.3 Uninstalling WDRReportCom

1. Double-click the **Add/Remove Programs** icon in the Windows Control Panel.
2. Do one of the following:
  - For Windows 2000, Windows XP and Windows 2003 Edition:  
Click **WDRReportCom** in the **Currently installed programs** box, and then click the **Change/Remove** button.
  - For Windows 98 and Windows NT 4.0:  
Click **WDRReportCom** on the **Install/Uninstall** tab, and then click the **Add/Remove** button.
3. Follow the instructions on the screen to complete uninstalling the program.

## 2.4 WDRReportCom.dll

WDRReportCom.dll is an ActiveX DLL that provides WDRReport object. You can write a program to work with the object. Before you can use the WDRReport object, you must create a reference to the object. And you should create references to Microsoft Word and Microsoft Graph Object Library too.

To create a reference to the WDRReport object

1. Do one of the following:
  - For Visual Basic 6.0  
From the **Project** menu, choose **References**.
  - For Microsoft Word Visual Basic For Application  
From the **Tools** menu, choose **References**.
2. In the **References** dialog box, select **WDRReportCom**.
3. You can use the **Browse** button to search for WDRReportCom.dll.
4. In the **References** dialog box, Select Microsoft Word and Microsoft Graph Object Library to create their references.

5. Declare an object variable of the object's class.

```
Dim wrpt As WDRReport
```

6. Assign an object reference to the variable by using the New keyword in a Set statement.

```
Set wrpt = New WDRReport
```

## 2.5 WordReport.exe

WordReport.exe is an executable program that developed using WDRReportCom.dll. It likes WDRReportGen command line and can read a WRF file to create a Word report. The syntax of command is:

```
wordreport <wrf file name> [-D] [-U1 user1] [-P1 pwd1] ... [-U10 user10]  
[-P10 pwd10] [pa1 pa2 ... pa10]
```

- |               |   |
|---------------|---|
| wrf file name | Specifying a WRF (.wrf) file that tells WDRReportCom how to get data from data sources and how to put data into a report.   |
| -D            | Display the generated report with Microsoft Word.   |
| -U1 user1 ... | Specify the user names. user1 is the user name of the first data source. user2 is the user name of the second data source.....  |
| -U10 user10   |   |
| -P1 pwd1 ...  | Specify the passwords. pwd1 is the password of the first data source. pwd2 is the password of the second data source.....   |
| -P10 pwd10    |   |
| pa1 ... pa10  | The values of the parameters defined in the WRF file. You can use parameters in SQL statements. WDRReportCom will replace the names of the parameters in a SQL statement with the actual values before it executes the SQL statement. You can use no more than 10 parameters in one report. |

For example, you have defined two parameters in your WRF file. The first parameter is the sales date, and the second is the category of the product.

You can run WordReport.exe as follows:

```
wordreport c:\wordreport\myreport.wrf 1996-05-01 "Dairy Products"
```

## 2.6 Run-Time Files

You can distribute royalty-free the run-time files of WDRReportCom with your applications. The run-time files are files your application must have in order to work correctly after installation. The following are the run-time files you need to distribute:

File	Description
wdreportcom.dll	The WDRReportCom ActiveX DLL. It must be registered.
wconv.cfg	The file contains the information of the file format. If you are using WordReport method to convert files, you should include it and copy it to the same directory as wdreportcom.dll.
sccrun.dll	Microsoft script runtime. WDRReportCom used some functions in this file. It should be copied to Windows System directory, and must be registered.

To register a DLL file, use regsvr32.exe. For example,  
regsvr32.exe /s "C:\Program Files\LJZsoft\WDRReportCom\WDRReportCom.dll"

## Chapter 3 Quick Start

### 3.1 Learning how to use WDRReportCom

You can teach yourself how to use WDRReportCom by choosing from the methods available in this section:

- You can study the samples included with WDRReportCom.
- You can use the detailed descriptions and instructions in this document.

### 3.2 Sample Database

WDRReportCom comes with Sample.mdb, a sample database you can use when learning the program. Sample.mdb is a Microsoft Access database. Virtually all of the examples in this manual are based on Sample.mdb data. The sample reports access the sample database through the ODBC data source name "Report Sample". When you install WDRReportCom, you can choose to add the ODBC data source name. And you also can add the ODBC data source name manually.

To create the System DSN "Report Sample", do as follows:

1. Click the Windows **Start** button, choose **Settings**, and then click **Control Panel**.
2. On computers running Microsoft Windows 2000 or later, double-click **Administrative Tools**, and then double-click **Data Sources (ODBC)**. The **ODBC Data Source Administrator** dialog box appears. On computers running previous versions of Microsoft Windows, double-click **32-bit ODBC** or **ODBC**.
3. Select the **System DSN** tab, and then press **Add** button.
4. Choose **Microsoft Access Driver (\*.mdb)**, then press **Finish** button.

5. In the **ODBC Microsoft Access Setup** dialog box, type **Report Sample** in the **Data Source Name** box.
6. Press the **Select** button, and browse to select **Sample.mdb**.
7. Press **OK** button to close the **ODBC Microsoft Access Setup** dialog box.
8. Press **OK** button to close the **ODBC Data Source Administrator** dialog box.

### 3.3 Samples

After WDRReportCom is installed, some samples are installed too. Use these samples to learn WDRReportCom.

The samples include a sample database, VB sample programs, VBA sample programs and sample reports. They are located in the Application Data\LJZsoft under All Users or your profile folder. WDRReportCom was tested with Microsoft Office 2007. Please download the sample reports for Microsoft Office 2007 from our website.

Directory	Description
{data}\Common\SampleDatabase	Contains the sample database "Sample.mdb".
{data}\WDRReportCom\Samples\WordReport	Contains the report template files (.doc) and the WRF files (.wrf).
{data}\WDRReportCom\Samples\VB	Contains the sample programs for VB6.0.
{data}\WDRReportCom\Samples\VBA	Contains the sample programs for Microsoft Word VBA.

{data} is the path of the data folder. You can select the data folder when you install WDRReportCom. By default, the data folder is the Application Data\LJZsoft folder under All Users. If you install WDRReportCom without administrative privileges, the data folder is the Application Data\LJZsoft folder under the current user. The data folder is usually at:

Windows 95/98: C:\windows\All Users\Application Data\LJZsoft

Windows NT: C:\WinNT\Profiles\All Users\Application Data\LJZsoft

Windows 2000/XP: C:\Documents and Settings\All Users\Application Data\LJZsoft

Windows Vista: C:\ProgramData\LJZsoft

### 3.4 Creating a Report Programmatically

#### 1. Create a template

In Microsoft Word, create a report template file named "custlist.doc". Static values and any Word features included in the template will be included in the generated report. The template file you have created as follows:

#### **Customer List**

Customer Name	City	Country	Contact Name

#### 2. Write the code in your application.

```
Set con = New ADODB.Connection
Set rec = New ADODB.Recordset
con.ConnectionString = "Data Source=Report Sample"
con.Open
strSQL = "SELECT CompanyName, CityName, CountryName,
ContactName FROM Customers, Cities, Countries WHERE
Customers.CityCode = Cities.CityCode AND Customers.CountryCode =
Cities.CountryCode AND Customers.CountryCode = Countries.CountryCode
ORDER BY CompanyName, CityName, CountryName"
rec.Open strSQL, con
wdrpt.VarTableReport Recordset:=rec, Table:=wdTable, CellList:="A2",
Reserve:=2
```

rec.Close

### 3.5 Creating a Report with WordReport.exe

#### 1. Create a template

In Microsoft Word, create a report template file named "custlist.doc". Static values and any Word features included in the template will be included in the generated report. The template file you have created as follows:

<b>Customer List</b>			
<b>Customer Name</b>	<b>City</b>	<b>Country</b>	<b>Contact Name</b>

#### 2. Create a WRF file

Create a WRF file named "custlist.wrf" using WDRReportGen or a text editor.

The following is the content of the WRF file.

WordReport Version 2.0

[Data Source]

Name1=Report Sample

[File]

ReportTemplateName=custlist.doc

ReportFileName=Report\custlist.doc

LogFileName=Log\custlist.log

[SQL]

@F1=Report(table=1 cell=A2 reserve=2)

SELECT CompanyName

,CityName

```
,CountryName  
,ContactName  
FROM Customers, Cities, Countries  
WHERE Customers.CityCode = Cities.CityCode  
AND Customers.CountryCode = Cities.CountryCode  
AND Customers.CountryCode = Countries.CountryCode  
ORDER BY CompanyName, CityName, CountryName
```

### 3. Run WordReport.exe

```
wordreport c:\report\custlist.wrf
```

# Chapter 4 Report Templates

## 4.1 About Reports

The report generated by WDRReportCom is a Microsoft Word document. The layouts, formats and styles of the report are defined by a report template, and the data of the report are got from databases such as Oracle, DB2.

## 4.2 About Report Templates

To make a report using WDRReportCom, you should create a report template first. The report template is a Microsoft Word document that defines the layouts, formats and styles of the report. In the Microsoft Word report template, you can input static content such as titles, descriptions, comments, a cover, a company logo, format the static content, and define the format of the data you will get from databases.

WDRReportCom will generate the report based on the report template file. All static contents and the layouts, formats and styles defined in the report template file will be brought to the final report file.

## 4.3 Word Basic Concepts

If you have known these concepts of Microsoft Word, please skip this section. For more detail information about Microsoft Word, refer to *Microsoft Word Help*.

### **4.3.1 Documents**

A document is a Microsoft Word file with extension .doc. You can open and save it using Microsoft Word. Microsoft Word documents may contain a combination of text, formatting and graphics.

### **4.3.2 Headers and Footers**

Headers and footers are areas in the top and bottom margins of each page in a document. You can insert text or graphics in headers and footers - for example, page numbers, the date, a company logo, the document's title or file name, or the author's name - that are printed at the top or bottom of each page in a document.

### **4.3.3 Tables**

A table is made up of rows and columns of cells that you can fill with text and graphics. Tables are often used to organize and present information. You can set borders, shading, alignment and fonts in tables.

### **4.3.4 Bookmarks**

A bookmark identifies a location or selection of text that you name and identify for future reference. For example, you might use a bookmark to identify text that you want to revise at a later time.

### **4.3.5 Page Breaks**

When you fill a page with text or graphics, Microsoft Word inserts an automatic page break and starts a new page. To force a page break at a specific location, you can insert a manual page break.

### **4.3.6 Graphics and Diagrams**

There are two basic types of graphics that you can use to enhance your Microsoft Word documents: drawing objects and pictures.

Drawing objects include AutoShapes, diagrams, curves, lines, and WordArt drawing objects. These objects are part of your Word document. Use the Drawing toolbar to change and enhance these objects with colors, patterns, borders, and other effects.

Pictures are graphics that were created from another file. They include bitmaps, scanned pictures and photographs, and clip art. You can change and enhance pictures by using the options on the Picture toolbar and a limited number of options on the Drawing toolbar. In some cases, you must ungroup and convert a picture to a drawing object before you can use the Drawing toolbar options.

### **4.3.7 Inline Pictures and Floating Pictures**

Inline picture: A graphic or other object that is positioned directly in the text of a Microsoft Word document at the insertion point.

Floating picture: A graphic or other object that is inserted in the drawing layer so that you can position it precisely on the page or in front of or behind text or other objects.

### **4.3.8 Charts**

Charts are visually appealing and make it easy for users to see comparisons, patterns, and trends in data. You can create a chart in a Microsoft Word document using Microsoft Graph or Microsoft Excel. When you create a new chart in Word, Microsoft Graph or Microsoft Excel opens and a chart is displayed with its associated data in a data sheet or worksheet.

### **4.3.9 Formatting**

You can use these formatting features of Microsoft Word to effectively display your data.

- **Characters formatting**

To make text stand out, you can format the text in selected characters. You can set font, color, size of text, bold and italic formats, animate or highlight the text.

- **Paragraphs formatting**

You can set text alignment, tab stops, line spacing, spacing before or after paragraphs, and borders.

- **Bulleted and numbered lists**

Bulleted and numbered lists in Microsoft Word are easy to create. You can quickly add bullets or numbers to existing lines of text, or Microsoft Word can automatically create lists as you type.

- **Borders, Shading, and Graphic Fills**

Borders, shading, and graphic fills can add interest and emphasis to various parts of your document. You can add borders to pages, text, tables and table cells, graphic objects, pictures, and Web frames. You can shade paragraphs and text. You can apply colored or textured fills to your graphic objects.

- **Automatic formatting**

By using AutoFormat, you can quickly apply formatting such as headings, bulleted and numbered lists, borders, numbers, symbols, and fractions to your text. You can automatically format a document either as you type or after you've written it. In both cases, you can control which automatic changes Microsoft Word makes. You can also turn off automatic formatting.

- **Style**

A style is a set of formatting characteristics that you can apply to text, tables,

and lists in your document to quickly change their appearance. When you apply a style, you apply a whole group of formats in one simple task. For example, instead of taking three separate steps to format your title as 16 pt, Arial, and center-aligned, you can achieve the same result in one step by applying the Title style.

### 4.3.10 Fields

#### ■ Fields

Fields are used as placeholders for data that might change in a document and for creating form letters and labels in mail-merge documents. Microsoft Word inserts fields when you use particular commands, such as the Date and Time command on the Insert menu. You can also manually insert your own fields by using the Field command on the Insert menu. Field codes appear between curly brackets, or braces ( { } ). Fields are somewhat like formulas in Microsoft Excel — the field code is like the formula, and the field result is like the value that the formula produces. You can switch between displaying field codes and results in your document.

#### ■ (Formula) fields

(Formula) field calculates a number by using a mathematical formula. You can insert an (Formula) field in a table or in regular text. Computation in tables can be completed using (Formula) fields, such as add, subtract, multiply, divide, sum. Syntax:

*{ = Formula [Bookmark ] [\# Numeric Picture ] }*

#### ■ DocVariable field

DocVariable field defines a document variable. Each document has a collection of variables, which can be added and referenced by the Microsoft Visual Basic for Applications programming language. This field provides a way to display the contents of the document variables in the document. Syntax:

`{ DOCVARIABLE "Name" }`

Inserts the string assigned to a document variable. "Name" is the name of the document variable.

## **4.4 Table Reports**

### **4.4.1 About Table Reports**

A table is made up of rows and columns of cells that you can fill with text and graphics. Tables are often used to make reports, and organize and present information.

WDRReportCom supports two types of table reports: fixed table report, variable table report.

Fixed table report: The number of rows and columns in the table is fixed.

When WDRReportCom executes a SQL statement, directly puts the result data into cells in the table.

Variable table report: The number of rows or columns in the table is unfixed, and it is variable as the number of result records. When WDRReportCom executes a SQL statement, it repeats the table rows or columns for each record or group, and then puts data into cells of the table.

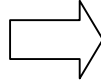
### **4.4.2 Creating a Table for a Fixed Table Report**

For a fixed table report, you need to create a table in the report template file according to the report. The format of the table is the same as the format in the report, but cells that should be filled data into are blank. When

WDRReportCom executes a SQL statement, the data values from data source will be filled into these cells.

	<b>A</b>	<b>B</b>
1		
2		
3		

The fixed table defined in the report template file



	<b>A</b>	<b>B</b>
1	14	3.4
2	20	5.2
3	8	2.7

The fixed table filled data by rows in the report file

### 4.4.3 Creating a Table for a Variable Table Report

For a variable table report, you also need to create a table in the report template file according to the report. But you just need to reserve some rows/columns in the table for one or two records. WDRReportCom will add some rows/columns according to the number of the records returned from data source.

<b>Date</b>	<b>Item Id</b>	<b>Sales</b>



<b>Date</b>	<b>Item Id</b>	<b>Sales</b>
1998-01-01	3	150
1998-01-02	3	200
1998-01-03	3	250
1998-01-05	3	350
1998-01-10	3	550
1998-01-21	3	150
1998-01-25	3	200
1998-01-31	3	100

The variable-rows table defined in the report template file

The variable-rows table filled data by rows in the report file

One record from data source can be put into two or more rows/columns. To do this, you need to create a repeat range that includes two or more rows/columns.

The format of the last row/column border can be different from the others. For

example, the outside borders used double lines, and the inside borders used single lines. To do this, you should reserve the blank rows/columns for 2 records. When WDRReportCom inserts some blank rows/columns, the new rows/columns will inherit the format of the first row/column in the reserved rows/columns.


WDRReportCom will repeat the range for each record. Ranges can be nested. The inside range is the detail range for detail record, and the external range is the group range for group. WDRReportCom will repeat the inside range for each record, and repeat the group range for each group.

#### **4.4.4 Formatting Cells**

To format cells that contain static contents, use “**Format**” menu in Microsoft Word. To format cells that contain (Formula) fields, use the switches of fields in Microsoft Word. For more detail information, refer to *Microsoft Word Help*. For cells in which data are got from database, you can set font, color, alignment using Microsoft Word. But to display values in formatting string, you should use other way.

You should write formatting expressions into data cells in the report template file. WDRReportCom will get the text of the cell as a format expression before it puts a value into a cell, and output the value using the format expression. In fact, WDRReportCom calls the format function in Visual Basic. The text got from a cell is used as the format expression in format function. For a variable table report, WDRReportCom will use the format expressions in the reserved rows/columns. For more information about format expression, refer to “Format Expression in Data Cells”.

Date	Quantity	Amount
yyyy-mm-dd	###0	###0.00
yyyy-mm-dd	###0	###0.00
yyyy-mm-dd	###0	###0.00



Date	Quantity	Amount
1999-02-18	560	827.79
1999-06-14	890	1,113.05
2000-01-21	1,240	1,552.25

The table defined in the report template file

The table generated in the report file

A format expression for numbers can have from one to four sections separated by semicolons. You can define the different formats and colors for positive values, negative values and zeros.

For example, the format "\$#,##0;(\$#,##0)" has two sections: the first defines the format and color (black) for positive values and zeros; the second section defines the format and color (red) for negative values. It displays "2345.12" as "\$2,345", displays "-5432" as "(\$5,432)".

The format "#,##0.00;;" has three sections: the first defines the format and color (black) for positive values, the second defines the format and color (red) for negative values, the third section defines the format and color (blue) for zeros. Note, the first semicolon ";" is red, the second semicolon ";" is blue. The negative values and zeros are printed using the format of the positive value. But the color for negative values is red, the color for zeros is blue. It displays "8.9" as "8.90", displays "-123" as "-123.00", and displays "0" as "0.00".

#### 4.4.5 Irregular Tables

Tables don't have to consist of simple grids. Not every row has to have the same number of columns. You can merge and split cells to create irregular tables. An irregular table is the table that contains split cells or merge cells, and it does not have the same number of cells for each row or column. While an irregular table provides for an attractive way to display data, but it does make it harder to process the document. You have some difficulty to reference

a cell in an irregular table. For example in the following table, for most Office version, cell1 is in column 3 and row 2, cell2 is in column 3 and row 3. But for some lower Office version, cell2 is in column 2 and row 3. Moreover, an error may occur when you try to work with some rows or columns in an irregular table.

A		B	C
		Cell1	
		Cell2	

Irregular table

To simplify your work and ensure that report function can be executed correctly, you should regularize the irregular tables. Split the merge cells, and remove the border in these cells. For example, the following table is a regularized table, cell1 is in column 3 and row 2, and cell2 is in column 3 and row 3.

A		B	C
		Cell1	
		Cell2	

Regularized table

#### 4.4.6 Referencing Cells

You can reference table cells as A1, A2, B1, B2, and so on, with the letter representing a column and the number representing a row. Cell references in Microsoft Word are always absolute references and are not shown with dollar signs. You can reference an entire row or column in a calculation in the following ways:

- Use a range that includes only the letter or number that represents it - for example, 1:1 to reference the first row in the table. This designation allows the calculation to automatically include all the cells in the row if you decide

to add other cells later.

- Use a range that includes the specific cells - for example, a1:a3 to reference a column with three rows. This designation allows the calculation to include only those particular cells. If you add other cells later and you want the calculation to include them, you need to edit the calculation.

Unfortunately, there is no inherent capability of Microsoft Word to inform you of the reference of a cell you have selected. You can get around this problem by using a macro. In the sample file “monthly\_sales.doc”, there is a macro called “CellRef”. The macro can tell you the table number and the cell reference. When you position the insertion pointer in a table cell and then run the macro, it displays a message box that shows the table number and the cell reference of the current cell. You can copy the macro “CellRef” into your documents. It is easy for you to reference a cell.

#### **4.4.7 Referencing Tables**

You can reference a table by an index number or a bookmark. The index number represents the position of the table in a document. The index number starts at 1. So table 1 is the first table in a document, table 2 is the second table, and so on. You can reference a nested table inside a table by an index number like 2-1-2. Table 2-1 is the first table inside table 2, and table 2-1-2 is the second table inside table 2-1. The max nested level WDRReportCom supports is 3. In the sample file “monthly\_sales.doc”, there is a macro called “CellRef”. The macro can tell you the index number of a table. When you position the insertion pointer in a table cell and then run the macro, it displays a message box that shows the table number and the cell reference of the current cell.

You can reference a table by the bookmark too. If you add a bookmark in a

table, you can reference the table using the bookmark. If you want to reference a nested table inside a table, you must add the bookmark in the nested table. It is very useful if you do not know the number of tables. To add a bookmark in a table, do as follows:

1. Click in the upper-left cell in the first row of the table, and place the insertion point before the text.
2. On the **Insert** menu, click **Bookmark**.
3. Under **Bookmark name**, type or select a name.

Bookmark names must begin with a letter and can contain numbers. You can't include spaces in a bookmark name.

4. Click **Add**.

Note: Microsoft Word 97 or lower does not support the nested tables.

#### **4.4.8 Formatting Cells for Pictures**

To enhance the visual impact of your report, you can insert pictures into your report. WDRReportCom supports many popular graphics file formats: bitmap, JPG, GIF, PNG, TIFF and so on. For the graphics file formats WDRReportCom supports, refer to *Microsoft Word Help*.

You should store the path and name of the graphics files in the database, and identify the image fields in the report function. WDRReportCom will read the graphics files, and insert them into the cells in the report file. You can adjust the position of the pictures by changing the cell margins.

To specify the inserted way, text wrapping style and size, you should write a formatting expression into the cell in the report template file. WDRReportCom will get the text of the cell, and insert a picture into the cell according to the instruction in the format expression. The format expression for pictures as follows:

[wrapstyle] [size]

The **wrapstyle** specifies the inserted way and text wrapping style, and can be one of the following values. The default value is NONE. "NONE" means floating pictures.

Values	Description
INLINE	Inline picture
NONE	Floating picture, none text-wrapping style
SQUARE	Square text-wrapping style
THROUGH	Through text-wrapping style
TIGHT	Tight text-wrapping style
TOPBOTTOM	TopBottom text-wrapping style

The **size** specifies the size of a picture. Possible values are STRETCH, Wnnn or / and Hnnn. "STRETCH" means that the picture is resized to fit within the cell. "W100" means that the width of the picture is set to 100 points. "H50" means that the height of the picture is set to 50 points. The default means the original size. If you just specify the width or height of the picture, not both, WDRReportCom will retain the original proportions of the picture when WDRReportCom resize it.

### Example

```
inline w120 h90
```

### Remarks

WDRReportCom will insert an inline picture, and set the width of the picture to 120 points, the height to 90 points.

Note: On Microsoft Word 97 or lower version, it may not work correctly if you insert a floating picture into a table. And it will split the table under some text wrapping style. You should insert inline pictures into a table.

## 4.5 Form Reports

### 4.5.1 About Form Reports

Beside table reports, WDRReportCom supports form reports too. For a form

report, you can get data from data sources, and put data as text, list, title and table in the report file. So you can make a form report as follows:

## **AUSTRALIA**

**Company:** G'day, Mate

**Contact:** Wendy Mackenzie      **Title:** Sales Representative

**Address:** 170 Prince Edward Parade Hunter's Hill

**City:** Sydney      **Country:** Australia

**Postal Code:** 2042

**Phone:** (02) 555-5914      **Fax:** (02) 555-4873

**Home Page:** <http://www.microsoft.com/accessdev/sampleapps/gdaymate.htm>

### **4.5.2 Creating Merge Fields or Quote Fields**

The report template file of a form report must contain merge fields or quote fields where data values will be inserted. A field can be created in text, list, title, table or other. When WDRReportCom is run, it will replace the merge fields or quote fields with values from data source.

To create a merge field or quote field:

1. Click where you want to insert a field.
2. On the **Insert** menu, click **Field**.
3. From the **Field names** list, select **MergeField** or **Quote**.
4. In the **Field name** text box, enter a name for the merge field or quote field.

For example, enter ProductName. Remember the field name, you will use it in the report function.

You can show field codes, and edit the field codes. To switch between field codes and results, do one of the following:

- To show or hide the field code for a specific field, click the field or the field results, and then press SHIFT+F9.
- To show or hide field codes for all fields in the document, press ALT+F9.

If the data value is a number, date or time, and you want to display it in a

custom format, use the switches of fields like “\#” or “\@”. For more detail information, refer to *Microsoft Word Help*.

### 4.5.3 Defining Ranges

A range represents a contiguous area in a document, and contains text, lists, tables, table rows or paragraphs. When WDRReportCom generates a report, it will repeat the range for each record or group.

A range can be defined by a Word bookmark. Or the entire document will be defined as the default range.

To define a range with a bookmark:

1. Create a document as your report template. For example, you create a template as follows:

Product Name	Product ID	Quantity Per Unit	Unit Price
«ProductName»	«ProductID»	«QuantityPerUnit»	«UnitPrice»

2. Select the range that you want to define as a repeat block. For example, you select a table row.

Product Name	Product ID	Quantity Per Unit	Unit Price
«ProductName»	«ProductID»	«QuantityPerUnit»	«UnitPrice»

3. On the **Insert** menu, click **Bookmark**.
4. Under **Bookmark name**, enter a bookmark name and click **Add**. For example, enter Product.

In Microsoft Word, bookmarks are hidden by default. To show bookmarks:

1. On the **Tools** menu, click **Options**, and then click the **View** tab.
2. Select the **Bookmarks** check box.
3. The bookmark appears in brackets ([...]) on the screen.

Product Name	Product ID	Quantity Per Unit	Unit Price
[«ProductName»]	«ProductID»	«QuantityPerUnit»	«UnitPrice»

Ranges can be nested. The inside range is for detail record, and the external

range is for group. For example, you defines a bookmark Category for the group of product category, and a bookmark Product for the detail record of product.

<b>«CategoryName»</b>			
«Description»			
Product Name	Product ID	Quantity Per Unit	Unit Price
«ProductName»	«ProductID»	«QuantityPerUnit»	«UnitPrice»

#### 4.5.4 Formatting Fields for Pictures

To enhance the visual impact of your report, you can insert pictures into your report. WDRReportCom supports many popular graphics file formats: bitmap, JPG, GIF, PNG, TIFF and so on. For the graphics file formats WDRReportCom supports, refer to *Microsoft Word Help*.

You should store the path and name of the graphics files in the database, and identify the image fields in the report function. WDRReportCom will read the graphics files, and insert them into the fields in the report file.

To specify the inserted way, text wrapping style and the size, you should write a formatting expression in the field switch “\#” in the report template file.

WDRReportCom will get the format string in the field switch “\#”, and insert a picture according to the instruction in the format expression. The format expression for pictures as follows:

[wrapstyle] [size]

The **wrapstyle** specifies the inserted way and text wrapping style, and can be one of the following values. The default value is INLINE. “INLINE” means inline pictures.

Values	Description
--------	-------------

INLINE	Inline picture
NONE	Floating picture, none text-wrapping style
SQUARE	Square text-wrapping style
THROUGH	Through text-wrapping style
TIGHT	Tight text-wrapping style
TOPBOTTOM	TopBottom text-wrapping style

The **size** specifies the size of a picture. Possible values are Wnnn or / and Hnnn. “W100” means that the width of the picture is set to 100 points. “H50” means that the height of the picture is set to 50 points. The default means the original size. If you just specify the width or height of the picture, not both, WDRReportCom will retain the original proportions of the picture when WDRReportCom resize it.

**Example**

`/# “square w84”`

**Remarks**

On the supposition that the original picture is size 144 x 168 points. WDRReportCom will insert a floating picture, apply the square text-wrapping style, set the height of the picture to 72 points, and the width to 84 points.

## 4.6 Charts

### 4.6.1 About Charts

You can create many different types of charts in Microsoft Word. The chart software may be Microsoft Graph or Microsoft Excel. It is depended on the version of your Microsoft Word. For Microsoft Word 2003 or earlier, the default chart software is Microsoft Graph. For Microsoft Word 2007 or later, the default chart software is Microsoft Excel.

WDRReportCom supports two kinds of charts created by Microsoft Graph or Excel. It executes a SQL statement, and puts the result data into the datasheet or worksheet of the chart. To work with charts created in Graph or

Excel, you must have Graph or Excel installed.

#### **4.6.2 Creating a Blank Chart using Microsoft Graph**

To create a Graph chart in the report using WDRReportCom, you need to add a Graph chart in the report template file first. The chart will be brought into the report file with the same chart type, display option, data format, label format and other chart item

If your Microsoft Office is earlier than Office 2007, or Microsoft Excel 2007 is not installed, when you create a new chart in Microsoft Word, Microsoft Graph opens.

To add a Graph chart in the template file:

1. Open the report template file using Microsoft Word.
2. On the **Insert** menu, click **Chart**.
3. Change the sample data on the datasheet as you need.
4. Modify the chart. For example, you want to change the chart type, make the text larger, or change colors, patterns, lines, fills, and borders in charts.
5. After you have finished the modification, delete data from the chart. You should keep a blank chart in the report template file. WDRReportCom will put data into the datasheet of the chart.
6. Change the chart to an inline shape if it is a floating shape. On the **Format** menu, click **Object**, click the **Layout** tab, and then click **In Line** text-wrapping style.

For more detail information, refer to *Microsoft Word Help* and *Microsoft Graph Help*.

#### **4.6.3 Creating a Blank Chart using Microsoft Excel**

To create an Excel chart in the report using WDRReportCom, you need to add an Excel chart object in the report template file first. The chart will be brought

into the report file with the same chart type, display option, data format, label format and other chart item.

To add an Excel chart in the template file:

1. Open the report template file using Microsoft Word.
2. Insert a chart with a chart sheet and a worksheet. For more information to insert an Excel chart object in Microsoft Word, please refer to the following part.
3. Change the sample data on the worksheet as you need.
4. Modify the chart. For example, you want to change the chart type, make the text larger, or change colors, patterns, lines, fills, and borders in charts.
  - If the report type is fix, the data range of the chart should be all rows/columns for the returned records.
  - If the report type is var, the data range of the chart should be 2 rows/columns.
5. After you have finished the modification, delete data from the chart. You should keep a blank chart in the report template file, and make the chart sheet active. WDRReportCom will put data into the worksheet of the chart.
6. Change the chart to an inline shape if it is a floating shape. On the **Format** menu, click **Object**, click the **Layout** tab, and then click **In Line** text-wrapping style.

By default, Microsoft Word 2007 uses Microsoft Excel to create charts, but doesn't expose the chart as a normal Excel object. There are two ways to create Excel chart object.

The first way for Microsoft Word 2007: Create a chart in Microsoft Excel, and copy it into Microsoft Word as an Excel chart object.

1. Open Microsoft Word and Microsoft Excel.
2. In Microsoft Excel, create a chart, and move the chart to a new worksheet.
3. In Microsoft Excel, click on the chart, and copy the chart.

4. Switch to Microsoft Word, place the insertion point where you want to create the chart.
5. On the **Home** tab, in the **Clipboard** group, click **Paste Special** and choose **Microsoft Office Excel Chart Object**. And click **OK** button.

The second way for Microsoft Word 2007: Insert an Excel worksheet in Microsoft Word, and create a chart in the Excel worksheet object.

1. In Microsoft Word, place the insertion point where you want to create the chart.
2. On the **Insert** tab, in the **Tables** group, click **Table**, and then click **Excel Spreadsheet**. You will see an Excel worksheet object.
3. Right-click the object, point to **Worksheet Object** on the shortcut menu, and choose **Open** from the submenu. Microsoft Excel will appear.
4. Create a chart in Microsoft Excel, and move the chart to a new worksheet.
5. When you've finished, choose **Close & Return** from the **File** menu.

For Microsoft Word 2003 or earlier, you can import Excel charts object.

1. In Microsoft Word, place the insertion point where you want to create the chart.
2. Click **Object** on the **Insert** menu, and then select the **Microsoft Excel Chart**.
3. You can work the Excel chart object by right-clicking the object, and pointing to **Worksheet Object** on the shortcut menu, and choosing **Open** from the submenu.
4. When you've finished, choose **Close & Return** from the **File** menu.

For more detail information, refer to *Microsoft Word Help and Microsoft Excel Help*.

#### **4.6.4 Referencing Charts**

You can reference a chart by an index number or a bookmark. The index

number represents the position of the chart in a document. The index number starts at 1. So chart 1 is the first chart in a document, chart 2 is the second chart, and so on.

You can reference a chart by the bookmark too. You might use a bookmark to identify a chart if you assigned the bookmark to the chart. To add a bookmark, do as follows:

1. Create a chart in the report template.
2. Select the chart you want a bookmark assigned to.
3. On the **Insert** menu, click **Bookmark**.
4. Under **Bookmark name**, enter a bookmark name and click **Add**. For example, enter Chart1.

In Microsoft Word, bookmarks are hidden by default. To show bookmarks:

1. On the **Tools** menu, click **Options**, and then click the **View** tab.
2. Select the **Bookmarks** check box.
3. The bookmark appears in brackets ([...]) on the screen.

# Chapter 5 API Reference

## 5.1 Objects

### 5.1.1 WDRReport Object

Represents the WDRReportCom. WDRReport is the main class for report generation using WDRReportCom.

#### Using the WDRReport Object

The following example creates a WDRReport object in another application and then generates a report using a WRF file.

```
Dim wdrpt As WDRReport
Set wdrpt = New WDRReport
wdrpt.WordReport wdApp, "customer_list.wrf"
```

## 5.2 Methods

### 5.2.1 FixTableReport Method

Generates a fixed table report based on a template. In a fixed table report, the number of rows and columns is fixed. WDRReportCom gets data from a recordset object, and directly fills data into the cells of a table.

#### Syntax

*object*.FixTableReport(**Recordset**, **Table**, **CellList**, **Range**, **FillOrder**, **ImageList**, **PageBreak**)

*object* **Required.** The object is the WDRReport object.

**Recordset** **Required.** An object variable that represents the ADODB.Recordset object to provides data. Before calling this method, please keep the current record position to the first record.

**Table** **Required.** An object variable that represents the Word.Table object to be filled data.

**CellList** **Required.** A string that represents the list of cells separated by the “,” character. For example, “A2,B2,B3,D2,D3”. The cells in the *CellList* should correspond to the data source fields in the recordset. The value of the first field is put into the first cell, and the value of the second field is put into the second cell .....

**Range** **Optional.** A string that indicates the range in the table to be used for the records. WDRReportCom will skip or repeat the range for each record. A range is composed of some rows or columns. You can reference a range of cells like “2:4” or “B:D”. The default range is the area that includes all cells for the records.

**FillOrder** **Optional.** An integer that indicates the order in which WDRReportCom fills data. If the value is zero, fills data by rows. Otherwise fills data by columns. Default is 0.

**ImageList** **Optional.** A string that indicates which data source fields are the picture files. The *ImageList* is the list of data source fields separated by the “,” character. You can identify a field using the name of field or the index number of field, but not simultaneously. In data source, you stored the path and file name of the picture, not the picture. The file path can be a relative path, an absolute path or a URL. If it is a relative path, the base path is the path of the document.

**PageBreak** **Optional.** A string that indicates the page breaks. The unit of page

length is r that means record. For example, “6r” or “6” means that WDRreportCom will insert a page break per 6 records. Default is “” that means no page break.

## Example

This example uses FixTableReport method to make the report “Top 5 Employees for Sales”.

1. Create the template in Microsoft Word.

Rank	Employee Name	Quantity	Amount	Percent of Total
1		#,##0	\$#,##0.00	0.00%
2		#,##0	\$#,##0.00	0.00%
3		#,##0	\$#,##0.00	0.00%
4		#,##0	\$#,##0.00	0.00%
5		#,##0	\$#,##0.00	0.00%

2. Write the code in your application.

```

Set con = New ADODB.Connection
Set rec = New ADODB.Recordset
con.ConnectionString = "Data Source=Report Sample"
con.Open
    strSQL = "SELECT TOP 5 e.FirstName + ' ' + e.LastName,
SUM(d.Quantity), Sum(d.UnitPrice * d.Quantity * (1-d.Discount)) AS
SalesAmount, SalesAmount / (SELECT amount FROM tmp_amount) FROM
Orders o, OrderDetails d, Products p, Employees e WHERE o.OrderID =
d.OrderID AND d.ProductID = p.ProductID AND o.EmployeeID =
e.EmployeeID AND YEAR(o.OrderDate) = 1996 AND MONTH(o.OrderDate) =
04 GROUP BY e.FirstName, e.LastName ORDER BY 3 DESC"
rec.Open strSQL, con
wdrpt.FixTableReport Recordset:=rec, Table:=wdTable, CellList:="B2"
rec.Close

```

3. Generate the report.

Rank	Employee Name	Quantity	Amount	Percent of Total
1	Nancy Davolio	467	\$24,827.45	23.68%
2	Laura Callahan	912	\$20,728.13	19.77%
3	Janet Leverling	578	\$16,360.12	15.60%
4	Andrew Fuller	558	\$13,937.64	13.29%
5	Margaret Peacock	481	\$8,298.45	7.91%

## 5.2.2 VarTableReport Method

Generates a variable table report based on a template. In a variable table report, the number of rows or columns in the table is unfixed, and it is variable as the number of the result records. WDRReportCom gets data from a recordset object, inserts some blank rows/columns or copy a range for each record, then fills data into the cells of a table.

### Syntax

*object*.VarTableReport(**Recordset**, **Table**, **CellList**, **Range**, **Copy**, **Reserve**, **FillOrder**, **ImageList**, **PageBreak**, **NoData**)

*object* Required. The object is the WDRReport object.

**Recordset** Required. An object variable that represents the ADODB.Recordset object to provides data. Before calling this method, please keep the current record position to the first record.

**Table** Required. An object variable that represents the Word.Table object to be filled data.

**CellList** Required. A string that represents the list of cells separated by the “,” character. For example, “A2,B2,B3,D2,D3”. The cells in the *CellList* should correspond to the data source fields in the recordset. The value of the first

field is put into the first cell, and the value of the second field is put into the second cell .....

**Range** Optional. A string that indicates the range in the table to be used for the records. WDRReportCom will repeat the range for each record. A range is composed of some rows or columns. You can reference a range of cells like "2:4" or "B:D". The default range is the area that includes all cells for the records.

**Copy** Optional. An integer that indicates whether WDRReportCom will copy the range for each record. If the value is zero, WDRReportCom will insert the blank rows/columns of the range for each record. Otherwise it will copy the source range and insert the copied range for each record. Note:

WDRReportCom will use clipboard. You can not copy and paste during report generating.

**Reserve** Optional. An integer that indicates the number of records for which you reserved some rows/columns in the report template for the report.

Possible values are 1 or 2. One means you reserve some rows/columns for one record, and two means some rows/columns for two records. Default is 1.

**FillOrder** Optional. An integer that indicates the order in which WDRReportCom fills data. If the value is zero, fills data by rows. Otherwise fills data by columns. Default is 0.

**ImageList** Optional. A string that indicates which data source fields are the picture files. The *ImageList* is the list of data source fields separated by the "," character. You can identify a field using the name of field or the index number of field, but not simultaneously. In data source, you stored the path and file name of the picture, not the picture. The file path can be a relative path, an absolute path or a URL. If it is a relative path, the base path is the path of the document.

**PageBreak** Optional. A string that indicates the page breaks. The unit of page


length is r that means record. For example, “6r” or “6” means that WDRReportCom will insert a page break per 6 records. Default is “” that means no page break.

**NoDataOptional.** An integer that represents an option when no data are returned from data source. If the value is 1, WDRReportCom will delete the range when no data are returned. If the value is 2, it will delete the table. Default is 0. It means to do nothing.

### Example

This example uses VarTableReport method to make the report “Mail Label”.

1. Create the template in Microsoft Word.

	XYZ Limited Co.	
	XYZ Building No.88 AAA Street BBB District	
	Beijing China, 100123	
<b>To:</b>		

2. Write the code in your application.

```
Set con = New ADODB.Connection
```

```
Set rec = New ADODB.Recordset
```

```
con.ConnectionString = "Data Source=Report Sample"
```

```
con.Open
```

```
strSQL = "SELECT CompanyName, Address, CityName & ', ' &
```

```
CountryName, PostalCode FROM Customers, Cities, Countries WHERE
```

```
Customers.CityCode = Cities.CityCode AND Customers.CountryCode =
```

```
Cities.CountryCode AND Customers.CountryCode = Countries.CountryCode
```

ORDER BY CompanyName"

rec.Open strSQL, con

wdrpt.VarTableReport Recordset:=rec, Table:=wdTable,

CellList:="B7,B8,B9,B10", Range:="1:11", Copy:=1, PageBreak:="4r"

rec.Close

### 3. Generate the report.

	XYZ Limited Co. XYZ Building No.88 AAA Street BBB District Beijing China, 100123	
<b>To:</b>	<b>Alfreds Futterkiste</b> Obere Str. 57 Berlin, Germany 12209	
	XYZ Limited Co. XYZ Building No.88 AAA Street BBB District Beijing China, 100123	
<b>To:</b>	<b>Ana Trujillo Emparedados y helados</b> Avda. de la Constitución 2222 México D.F., Mexico 05021	

### 5.2.3 GroupTableReport Method

Generates a variable table report based on a template, and groups data in the report. In a variable table report, the number of rows or columns in the table is unfixed, and it is variable as the number of the result records. WDRptCom gets data from a recordset object, copy the group range for each group, and copy the detail range for each record.

## Syntax

***object.GroupTableReport(Recordset, Table, CellList, Range, Copy, FillOrder, ImageList, PageBreak, NoData, Group1, GroupRange1, ... Group10, GroupRange10)***

*object* Required. The object is the WDRReport object.

**Recordset** Required. An object variable that represents the ADODB.Recordset object to provides data. Before calling this method, please keep the current record position to the first record.

**Table** Required. An object variable that represents the Word.Table object to be filled data.

**CellList** Required. A string that represents the list of cells separated by the “,” character. For example, “A2,B2,B3,D2,D3”. The cells in the *CellList* should correspond to the data source fields in the recordset. The value of the first field is put into the first cell, and the value of the second field is put into the second cell .....

**Range** Optional. A string that indicates the range in the table to be used for the records. WDRReportCom will repeat the range for each record. A range is composed of some rows or columns. You can reference a range of cells like “2:4” or “B:D”. The default range is the area that includes all cells for the records.

**Copy** Optional. An integer that indicates whether WDRReportCom will copy the range for each record. If the value is zero, WDRReportCom will insert the blank rows/columns of the range for each record. Otherwise it will copy the source range and insert the copied range for each record. In general, inserting rows/columns is faster than copying range.

**FillOrder** Optional. An integer that indicates the order in which WDRReportCom fills data. If the value is zero, fills data by rows. Otherwise fills

data by columns. Default is 0.

**ImageList** Optional. A string that indicates which data source fields are the picture files. The *ImageList* is the list of data source fields separated by the “,” character. You can identify a field using the name of field or the index number of field, but not simultaneously. In data source, you stored the path and file name of the picture, not the picture. The file path can be a relative path, an absolute path or a URL. If it is a relative path, the base path is the path of the document.

**PageBreak** Optional. A string that indicates the page breaks. The unit of page length is r or g. "r" means record, "g1" means group one, "g2" means group two..... For example, “6r” or “6” means that WDRReportCom will insert a page break per 6 records, “1g1” or “1g” means a page break per group one, and “1g1,6r” means a page break per group one or 6 records. Default is “” that means no page break.

**NoData** Optional. An integer that represents an option when no data are returned from data source. If the value is 1, WDRReportCom will delete the range when no data are returned. If the value is 2, it will delete the table. Default is 0. It means to do nothing.

**Group1...Group10** Optional. A string that indicates the group that is the list of data source fields separated by the “,” character. You can identify a field using the name of field or the index number of field, but not simultaneously. In one report, there may be up to 10 groups. Notes: the order of groups should be in accordance with the order of ORDER BY clause in the SQL statement.

**GroupRange1...GroupRange10** Optional. A string that indicates the range of the group in the table. WDRReportCom will repeat the range for each group. The range of the group should contain the range of the details and the area that includes all cells for this group. You reference a group range like “2:4” or “B:D”. For example, there are two groups, the range of the group one contains

all cells for the group one and the range of the group two, and the range of the group two contains all cells for the group two and the range of the details. The default range is the area that includes all cells for this group and the range or the group range for the lower level group.

### Remarks

GroupTableReport method will use clipboard. You can not copy and paste during report generating.

### Example

This example uses GroupTableReport method to make the report "Customer Profile".

1. Create the template in Microsoft Word.

Customer Name	Contact Name	Phone/Fax	Address

2. Write the code in your application.

```

Set con = New ADODB.Connection
Set rec = New ADODB.Recordset
con.ConnectionString = "Data Source=Report Sample"
con.Open
strSQL = "SELECT LEFT(CompanyName,1), CompanyName,
ContactName, 'Phone: ' & Phone, 'Fax: ' & Fax, Address, CityName & ', ' &
CountryName, PostalCode FROM Customers, Cities, Countries WHERE
Customers.CityCode = Cities.CityCode AND Customers.CountryCode =
Cities.CountryCode AND Customers.CountryCode = Countries.CountryCode
ORDER BY CompanyName"
rec.Open strSQL, con

```

```

wdrpt.GroupTableReport Recordset:=rec, Table:=wdTable,
CellList:="A2,B3,C3,D3,D4,E3,E4,E5", Copy:=1, Range:="2:5", Group1:="1",
PageBreak:="5r"
rec.Close

```

### 3. Generate the report.

	Customer Name	Contact Name	Phone/Fax	Address
<b>A</b>				
	Alfreds Futterkiste	Maria Anders	Phone: 030-0074321 Fax: 030-0076545	Obere Str. 57 Berlin, Germany 12209
	Ana Trujillo Emparedados y helados	Ana Trujillo	Phone: (5) 555-4729 Fax: (5) 555-3745	Avda. de la Constitución 2222 México D.F., Mexico 05021
<b>B</b>				
	Berglunds snabbköp	Christina Berglund	Phone: 0921-12 34 65 Fax: 0921-12 34 67	Berguvsvägen 8 Luleå, Sweden S-951 22
	Blauer See Delikatessen	Hanna Moos	Phone: 0621-08460 Fax: 0621-08924	Forsterstr. 57 Mannheim, Germany 68306

## 5.2.4 FormReport Method

Generates a form report based on a template, and groups data in the report.

For a form report, you can put data from data source as text, list, title and table in the report file. WDRReportCom gets data from a recordset object, copy the group range for each group, and copy the detail range for each record.

### Syntax

*object*.FormReport(**Recordset**, **Document**, **CellList**, **Range**, **ImageList**, **PageBreak**, **NoData**, **Group1**, **GroupRange1**, ... **Group10**, **GroupRange10**)

**Recordset** Required. An object variable that represents the ADODB.Recordset object to provides data. Before calling this method, please

keep the current record position to the first record.

**Document** Required. An object variable that represents the Word.Document object to be filled data.

**CellList** Required. A string that represents the list of merge fields or quote fields separated by the “,” character. For example, “ProductName,ProductID,QuantityPerUnit,UnitPrice”. The merge fields or quote fields in the *celllist* should correspond to the data source fields in the SQL statement. The value of the first data source field is put into the first merge field or quote field, and the value of the second data source field is put into the second merge field or quote field .....

**Range** Optional. A string that indicates the range to be used for the records. WDRReportCom will repeat the range for each record. A range is defined by a bookmark. You reference a range using a bookmark name. The default range is the group range or the entire document.

**ImageList** Optional. A string that indicates which data source fields are the picture files. The *ImageList* is the list of data source fields separated by the “,” character. You can identify a field using the name of field or the index number of field, but not simultaneously. In data source, you stored the path and file name of the picture, not the picture. The file path can be a relative path, an absolute path or a URL. If it is a relative path, the base path is the path of the table.

**PageBreak**Optional. A string that indicates the page breaks. The unit of page length is r or g. "r" means record, "g1" means group one, "g2" means group two..... For example, “6r” or “6” means that WDRReportCom will insert a page break per 6 records, “1g1” or “1g” means a page break per group one, and “1g1,6r” means a page break per group one or 6 records. Default is “” that means no page break.

**NoData**Optional. An integer that represents an option when no data are

returned from data source. If the value is not zero, WDRReportCom will delete the range when no data are returned. Default is 0.

**Group1...Group10** Optional. A string that indicates the group that is the list of data source fields separated by the “,” character. You can identify a field using the name of field or the index number of field, but not simultaneously. In one report, there may be up to 10 groups. Notes: the order of groups should be in accordance with the order of ORDER BY clause in the SQL statement.

**GroupRange1...GroupRange10** Optional. A string that indicates the range of the group. WDRReportCom will repeat the range for each group. A range is defined by a bookmark. You reference a range using a bookmark name. The range of the group should contain the range of the details and the area that includes all merge fields or quote fields for this group. For example, there are two groups, the range of the group one contains all merge fields or quote fields for the group one and the range of the group two, and the range of the group two contains all merge fields or quote fields for the group two and the range of the details. The default range is the range of the upper level group or the entire document.

### **Remarks**

FormReport method will use clipboard. You can not copy and paste during report generating.

### **Example**

This example uses FormReport method to make the report “Product Catalog”.

1. Create the template in Microsoft Word.

«CategoryName»

«Description»

Product Name	Product ID	Quantity Per Unit	Unit Price
«ProductName»	«ProductID»	«QuantityPerUnit»	«UnitPrice»

2. Write the code in your application.

```
Set con = New ADODB.Connection
Set rec = New ADODB.Recordset
con.ConnectionString = "Data Source=Report Sample"
con.Open
strSQL = " SELECT CategoryName, Description, ProductName,
ProductID, QuantityPerUnit, UnitPrice FROM Products, Categories WHERE
Products.CategoryID = Categories.CategoryID ORDER BY 1,3"
rec.Open strSQL, con
wdrpt.FormReport Recordset:=rec, Document:=wdDoc,
CellList:="CategoryName, Description, ProductName, ProductID,
QuantityPerUnit, UnitPrice", Range:="Product", Group1:="1,2",
GroupRange1:="Category"
rec.Close
```

3. Generate the report.

## Beverages

Soft drinks, coffees, teas, beers, and ales

Product Name	Product ID	Quantity Per Unit	Unit Price
Chai	1	10 boxes x 20 bags	\$18.00
Chang	2	24 - 12 oz bottles	\$19.00
Chartreuse verte	39	750 cc per bottle	\$18.00
Côte de Blaye	38	12 - 75 cl bottles	\$263.50
Guaraná Fantástica	24	12 - 355 ml cans	\$4.50
Ipoh Coffee	43	16 - 500 g tins	\$46.00

### 5.2.5 MSGraphChart Method

Generates a chart based on a template using Microsoft Graph.

WDRReportCom gets data from a recordset object, and fills data into the datasheet of a chart in the report file.

#### Syntax

*object*. **MSGraphChart(Recordset, Chart, CellList, Range, FillOrder)**

**Recordset** Required. An object variable that represents the ADODB.Recordset object to provides data. Before calling this method, please keep the current record position to the first record.

**Chart** Required. An object variable that represents the Graph.Chart object.

**CellList** Required. A string that represents the list of cells separated by the “,” character. For example, “A2,B2,B3,D2,D3”. The cells in the *CellList* should correspond to the data source fields in the recordset. The value of the first field is put into the first cell, and the value of the second field is put into the second cell .....Note: On the datasheet, the leftmost column and the top row,

which are commonly used for legend text or axis labels, are referred to as column 0 (zero) and row 0 (zero).

**Range** Optional. A string that indicates the range in the datasheet to be used for the records. WDRReportCom will skip the rows/columns of the range for each record. A range is composed of some rows or columns. You can reference a range of cells like "2:4" or "B:D". The default range is the area that includes all cells for the records.

**FillOrder** Optional. An integer that indicates the order in which WDRReportCom fills data. If the value is zero, fills data by rows. Otherwise fills data by columns. Default is 1.

### Example

This example uses MSGraphChart method to make the report "Sales by Categories".

1. Create the template in Microsoft Word.

The datasheet of the chart defined in the report template:

		A	B	C	D
	Category Name				
1	Amount				
2					

2. Write the code in your application.

```
Set con = New ADODB.Connection
```

```
Set rec = New ADODB.Recordset
```

```
con.ConnectionString = "Data Source=Report Sample"
```

```
con.Open
```

```
strSQL = " SELECT c.CategoryName, Sum(d.UnitPrice * d.Quantity *  
(1-d.Discount)) FROM Orders o, OrderDetails d, Products p, Categories c  
WHERE o.OrderID = d.OrderID AND d.ProductID = p.ProductID AND
```

```
p.CategoryID = c.CategoryID AND YEAR(o.OrderDate) = 1996 AND
MONTH(o.OrderDate) = 04 GROUP BY c.CategoryName ORDER BY
c.CategoryName"
```

```
rec.Open strSQL, con
```

```
wdrpt.Chart Recordset:=rec, Chart:=wdChart, CellList:="A0"
```

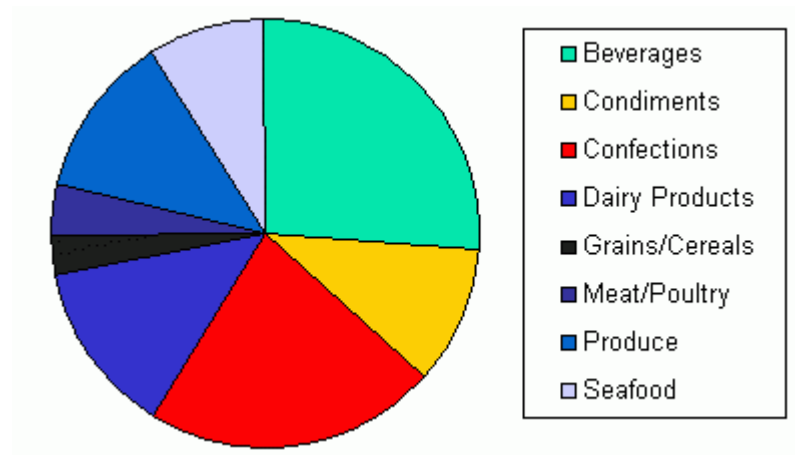
```
rec.Close
```

3. Generate the chart.

The datasheet of the chart generated in the report:

		A	B	C	D
	Category Name	Beverages	Condiments	Confections	Dairy Products
1	Amount	27761.57	10773.27	22877.18	13685.32
2					

The chart generated in the report:



## 5.2.6 ExcelChart Method

Generates a chart based on a template using Microsoft Excel. WDRReportCom gets data from a recordset object, and fills data into the worksheet of a chart in the report file.

### Syntax

*object*. **ExcelChart(Recordset, Workbook, CellList, ReportType, Range,**

## **FillOrder)**

**Recordset** Required. An object variable that represents the ADODB.Recordset object to provides data. Before calling this method, please keep the current record position to the first record.

**Workbook** Required. An object variable that represents the Excel.Workbook object.

**CellList** Required. A string that represents the list of cells separated by the “,” character. For example, “A2,B2,B3,D2,D3”. The cells in the *CellList* should correspond to the data source fields in the recordset. The value of the first field is put into the first cell, and the value of the second field is put into the second cell .....

**ReportType** Optional. An integer that indicates the report type. If the value is zero, WDRReportCom will add some blank rows/columns before filling data values into the worksheet of the chart. If the value is one, WDRReportCom will directly fill data vales into the worksheet of the chart. Default is 0. When it is a variable table report, you should reserve two rows/columns in the worksheet in the report template, and set the data range of the chart to 2 rows/columns.

**Range** Optional. A string that indicates the range in the worksheet to be used for the records. WDRReportCom will skip the rows/columns of the range for each record. A range is composed of some rows or columns. You can reference a range of cells like “2:4” or “B:D”. The default range is the area that includes all cells for the records.

**FillOrder** Optional. An integer that indicates the order in which WDRReportCom fills data. If the value is zero, fills data by rows. Otherwise fills data by columns. Default is 0.

## **Example**

This example uses ExcelChart method to make the report "Sales by Categories".

1. Create the template in Microsoft Word.

The worksheet of the chart defined in the report template:

	A	B	C	D	E	F
1	Category Name	Amount				
2						
3						
4						

2. Write the code in your application.

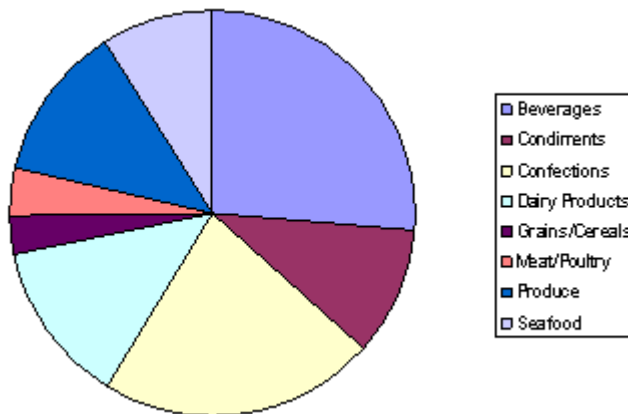
```
Set con = New ADODB.Connection
Set rec = New ADODB.Recordset
con.ConnectionString = "Data Source=Report Sample"
con.Open
strSQL = " SELECT c.CategoryName, Sum(d.UnitPrice * d.Quantity *
(1-d.Discount)) FROM Orders o, OrderDetails d, Products p, Categories c
WHERE o.OrderID = d.OrderID AND d.ProductID = p.ProductID AND
p.CategoryID = c.CategoryID AND YEAR(o.OrderDate) = 1996 AND
MONTH(o.OrderDate) = 04 GROUP BY c.CategoryName ORDER BY
c.CategoryName"
rec.Open strSQL, con
wdrpt.Chart Recordset:=rec, Workbook:=wdWorkbook, CellList:="A2"
rec.Close
```

3. Generate the chart.

The worksheet of the chart generated in the report:

	A	B	C	D	E	F
1	<b>Category Name</b>	<b>Amount</b>				
2	Beverages	27761.57499				
3	Condiments	10773.26999				
4	Confections	22877.17998				
5	Dairy Products	13685.32499				
6	Grains/Cereals	3325.399995				
7	Meat/Poultry	4083.659998				
8	Produce	13031.2				
9	Seafood	9316.544988				

The chart generated in the report:



## 5.2.7 SetDocVariable Method

Gets data from a recordset object, and assigns the values to the names defined in the Word document. WDRReportCom will just fetch the first record, no matter how many records are returned from data source.

### Syntax

*object*.SetDocVariable(**Recordset**, **Document**, **NameList**)

*object* **Required**. The object is the WDRReport object.

**Recordset** **Required**. An object variable that represents the ADODB.Recordset object to provides data.

**Document** **Required**. An object variable that represents the Word.Document object to be filled data.

**NameList** Required. A string that represents the name of the document variables you want assign values to. The *NameList* is the list of variable names separated by the “,” character. For example, “BeginDate, EndDate” means two document variables: BeginDate and EndDate that should be defined in the report template. The variables in the list should correspond to the fields in the SQL statement. The value of the first field is put into the first variable, and the value of the second field is put into the second variable...

### Remarks

SetDocVariable method supports headers and footers. You can use it to put data into headers or footers.

WDRReportCom can not update the docvariable fields automatically. To show the results, you should update the fields. For example,

‘ Update all fields in the document

```
wdDoc.Fields.Update
```

‘ Update all fields in the primary header of section 1.

```
wdDoc.Sections(1).Headers(wdHeaderFooterPrimary).Range.Fields.Update
```

### Example

This example uses SetDocVariable method to assign the values of fields to document variables.

1. Define the document variables BeginDate and EndDate in the report template in Microsoft Word.

2. Write the code in your application.

```
Set con = New ADODB.Connection
```

```
Set rec = New ADODB.Recordset
```

```
con.ConnectionString = "Data Source=Report Sample"
```

```
con.Open
```

```
strSQL = " SELECT min_date, max_date FROM tmp0"
```

```
rec.Open strSQL, con
wdrpt.SetDocVariable Recordset:=rec, Document:=wdDoc,
NameList:="BeginDate,EndDate"
rec.Close
wdDoc.Fields.Update
```

## 5.2.8 WordReport Method

Generates the reports based on the templates and a WRF file. The WRF file tells WDRptCom how to get data from data sources and how to put data into the reports.

### Syntax

*object*.WordReport(**Application**, **WrfFile**, **Param1 ... Param10**)

*object* *Required*. The object is the WDRptCom object.

**Application** *Required*. An object variable that represents the Word.Application object.

**WrfFile** *Required*. A string that represents the WRF file. You can include a full path.

**Param1 ... Param10** *Optional*. A string that represents the parameters. These parameters have been defined in the WRF file.

### Example

This example uses WordReport method to make the report "Customer List".

1. Create the template customer\_list.doc using Microsoft Word.
2. Create the WRF file customer\_list.wrf using a text editor.
3. Write the code in your application.

```
Set wdApp = New Word.Application
Set wdrpt = New WDRReport
Call wdrpt.WordReport(wdApp, "customer_list.wrf")
```

## 5.2.9 GetTableByIndex Method

Returns a Word.Table object by using the table index.

### Syntax

*object*.**GetTableByIndex**(**Document**, **Index**)

*object* *Required*. The object is the WDRReport object.

**Document** *Required*. An object variable that represents the Word.Document object.

**Index** *Required*. A string that represents the index of the table. The index number starts at 1. For examples, table 2 is the second table in the document. The index number of a nested table likes 2-1-2. For examples, table 2-1 is the first table inside table 2, and table 2-1-2 is the second table inside table 2-1. The max nested level WDRReportCom supports is 3.

### Example

This example uses GetTableByIndex method to get the first table in the document.

```
Dim wdDoc As Word.Document
Dim wdTable As Word.Table
.....
Set wdTable = mwdrpt.GetTableByIndex(wdDoc, "1")
```

### 5.2.10 GetTableByBookmark Method

Returns a Word.Table object by using the bookmark.

#### Syntax

*object*.GetTableByBookmark(**Document**, **Bookmark**)

*object* Required. The object is the WDRReport object.

**Document** Required. An object variable that represents the Word.Document object.

**Bookmark** Required. A string that represents the name of the bookmark that is in the table. You can reference a nested table. The max nested level WDRReportCom supports is 3.

#### Example

This example uses GetTableByBookmark method to get the table in the document.

```
Dim wdDoc As Word.Document
```

```
Dim wdTable As Word.Table
```

```
.....
```

```
Set wdTable = mwdrpt.GetTableByBookmark (wdDoc, "MyTable")
```

### 5.2.11 GetChartByIndex Method

Returns a Word.OLEFormat object that represents the Microsoft Graph chart or Microsoft Excel chart OLE object.

#### Syntax

*object*.GetChartByIndex(**Document**, **Index**)

*object* Required. The object is the WDRReport object.

**Document** Required. An object variable that represents the Word.Document object.

**Index** Required. A string that represents the index of the chart. The index number starts at 1. For examples, chart 2 is the second chart in the document.

### **Example**

This example uses GetChartByIndex method to get the object of the first chart in the document.

```
Dim wdDoc As Word.Document
Dim wdChartOLE As Word.OLEFormat
Dim graChart As Graph.Chart
Dim xlWorkbook As Excel.Workbook
.....
Set wdChartOLE = mwdrpt.GetChartByIndex(wdDoc, "1")
If Left(wdChartOLE.ProgId, 11) = "Excel.Chart" Then
    Set xlWorkbook = wdChartOLE.object
Else
    Set graChart = wdChartOLE.object
End If
```

### **5.2.12 GetChartByBookmark Method**

Returns a Word.OLEFormat object that represents the Microsoft Graph chart or Microsoft Excel chart OLE object.

#### **Syntax**

*object*.**GetChartByIndex**(**Document**, **Bookmark**)

*object Required.* The object is the WDRReport object.

**Document** Required. An object variable that represents the Word.Document object.

**Bookmark** Required. A string that represents the name of the bookmark that includes the chart.

### **Example**

This example uses GetChartByBookmark method to get the object of the chart in the document.

```
Dim wdDoc As Word.Document
Dim wdChartOLE As Word.OLEFormat
Dim graChart As Graph.Chart
Dim xlWorkbook As Excel.Workbook
.....
Set wdChartOLE = mwdrpt.GetChartByBookmark(wdDoc, "Chart1")
If Left(wdChartOLE.ProgId, 11) = "Excel.Chart" Then
    Set xlWorkbook = wdChartOLE.object
Else
    Set graChart = wdChartOLE.object
End If
```

## **5.3 Events**

### **5.3.1 BeforeConnect Event**

Occurs before a connection starts.

#### **Syntax**

**Private Sub *object*\_BeforeConnect(*UserID* As String, *Password* As String, *DataSource* As String, *Connection* As ADODB.Connection)**

*object* The object is the WDRReport object.

*UserID* A string that represents a user name for the connection.

*Password* A string that represents a password for the connection.

*DataSource* A string that represents a data source name for the connection.

*Connection* The ADODB.Connection object.

### **Example**

```
Private Sub mwdrpt_BeforeConnect(UserID As String, Password As String,
DataSource As String, Connection As ADODB.Connection)
```

```
    Connection.ConnectionTimeout = 15
```

```
    Connection.CursorLocation = adUseClient
```

```
End Sub
```

### **5.3.2 TemplateOpen Event**

Occurs when a template document is opened.

#### **Syntax**

**Private Sub *object*\_TemplateOpen(*ByVal Document* As Word.Document)**

*object* The object is the WDRReport object.

*Document* An object variable that represents the Word.Document object to be opened.

#### **Example**

```
Private Sub mwdrpt_TemplateOpen(ByVal Document As Word.Document)
    gstrFileName = Document.FullName
End Sub
```

### 5.3.3 ReportComplete Event

Occurs when all report generating process is completed.

#### Syntax

**Private Sub *object\_ReportComplete*(ByVal *Document* As Word.Document)**

*object* The object is the WDRReport object.

*Document* An object variable that represents the Word.Document object.

#### Example

```
Private Sub mwdrpt_ReportComplete(ByVal Document As Word.Document)
    ' Close the document and do not display the report when get errors
    If mintErrCount > 0 Then
        Document.Close
    End If
End Sub
```

### 5.3.4 FunctionBeforeExectue Event

Occurs before a function is executed.

#### Syntax

**Private Sub *object\_FunctionBeforeExectue*(ByVal *FunctionNo* As String,  
ByVal *FunctionType* As Integer, ByVal *SQLNo* As Long, ByVal *SQLText* As  
String)**

*object* The object is the WDRReport object.

*FunctionNo* A string that represents the label of the function.

*FunctionTyp* An integer that represents the type of the function. 0 means ExecSQL function. 1 means DocVariable function. 2 means Report function. 3 means Chart function.

*SQLNo* A long that represents the number of SQL statements.

*SQLText* A string that contains the SQL statement.

### **Example**

```
Private Sub mwd rpt_FunctionBeforeExectue(ByVal FunctionNo As String,  
ByVal FunctionType As Integer, ByVal SQLNo As Long, ByVal SQLText As  
String)
```

```
    frmWait.lblFunctionNo = FunctionNo
```

```
    frmWait.lblSQLCount = SQLNo
```

```
End Sub
```

### **5.3.5 FunctionAfterExectue Event**

Occurs after a function is executed.

#### **Syntax**

```
Private Sub object_FunctionAfterExectue(ByVal FunctionNo As String,  
ByVal FunctionType As Integer, ByVal SQLNo As Long, ByVal ErrObj As  
ErrObject)
```

*object* The object is the WDRReport object.

*FunctionNo* A string that represents the label of the function.

*FunctionTyp* An integer that represents the type of the function. 0 means

ExecSQL function. 1 means DocVariable function. 2 means Report function. 3 means Chart function.

*SQLNo* A long that represents the number of SQL statements.

*ErrObj* The Err object.

### **Example**

```
Private Sub mwd rpt_FunctionAfterExectue(ByVal FunctionNo As String,
ByVal FunctionType As Integer, ByVal SQLNo As Long, ByVal ErrObj As
ErrObject)
    If ErrObj.Number <> 0 Then
        If FunctionType <> 0 Then           'Ignore errors of EXECSQL
            MsgBox ErrObj.Description, vbExclamation, App.ProductName
            mintErrCount = mintErrCount + 1
        End If
    End If
End Sub
```

### **5.3.6 FunctionProgress Event**

Occurs periodically during a function processing.

#### **Syntax**

```
Private Sub object_FunctionProgress(ByVal Progress As Long, ByVal
RecordCount As Long)
```

*object* The object is the WDRreport object.

*Progress* A long that indicates the number of records that have currently been processed.

*RecordCount* A long that indicates the total number of records.

## Example

```
Private Sub mwdrpt_FunctionProgress(ByVal Progress As Long, ByVal  
RecordCount As Long)
```

```
    frmWait.IblRecordCnt.Caption = Format(Progress, "#,##0") & " / " &  
    Format(RecordCount, "#,##0")
```

```
End Sub
```

## 5.4 Error Messages

The following table lists the trappable errors for the WDRReport Object.

Value	Description
-2147221493	The file <i>WrfFileName</i> does not exist.
-2147221492	The file <i>WrfFileName</i> is not a WordReport file.
-2147221491	Error in reading the file <i>WrfFileName</i> ."
-2147221490	Report template file <i>TemplateFileName</i> does not exist.
-2147221489	The report file is not named correctly.
-2147221488	Failed to create the report file <i>ReportFileName</i> .
-2147221487	Failed to open the template file <i>TemplateFileName</i> .
-2147221486	Failed to save the report file.
-2147221485	Failed to save the report file. Not support the file format: <i>FileFormat</i> .
-2147221484	Failed to update Word fields.
-2147221473	The ADODB.Recordset object is closed.
-2147221453	Syntax error. The table <i>Table</i> does not exist.
-2147221452	Syntax error. The chart <i>Chart</i> does not exist.
-2147221451	The Word.Document object is not set.
-2147221450	The Word.Table object is not set.
-2147221449	The Graph.Chart object is not set.
-2147221448	The Excel.Workbook object is not set.
-2147221447	Unable to find the Word document.
-2147221446	Unable to find the Excel chart.
-2147221445	Unable to find the source data worksheet.
-2147221403	Syntax error. There is a lack of the parameter "NAME".
-2147221393	Syntax error. There is a lack of the parameter "CELL".
-2147221392	Syntax error. It is not a valid cell "" for the parameter "CELL".
-2147221391	Syntax error. Failed to parse cell list.
-2147221390	Syntax error. Failed to parse cell <i>Cell</i> .
-2147221389	Can not find the field <i>Field</i> in the range <i>Range</i> .
-2147221383	The range or copyrange should be <i>Range</i> .

-2147221382	Syntax error. Failed to parse range <i>Range</i> .
-2147221373	Syntax error. Failed to parse image. Can not find field <i>ImageField</i> in the image list.
-2147221372	Syntax error. Failed to parse image.
-2147221363	Syntax error. Failed to parse group <i>Group</i> . Can not find field <i>Field</i> .
-2147221362	Syntax error. Failed to parse group.
-2147221361	The grouprange of group <i>N</i> should be <i>Range</i> .
-2147221360	Syntax error. Failed to parse grouprange.

## Chapter 6 WRF Files

### 6.1 Using WRF files

#### 6.1.1 About WRF files

Like WDRptGen, WDRptCom also can read a WRF file to generate a report. The WRF file is a text file with a .wrf extension. It contains information such as the name of the report template file, the name of the report file, the name of the log file, data sources, parameters and functions. The WRF file tells WDRptCom how to get data from data sources and how to put data into a report. Using the WRF file, it will simplify your development.

#### 6.1.2 Using a WRF file with WDRpt Object

WDRpt object provides the WordReport method to generate a report based on a WRF file. For example, you have created the WRF file "myreport.wrf" and the template file. In the WRF file, there are two parameters. The first parameter is the sales date "\$SalesDate", and the second is the category of the products "\$Category". You can call WordReport method to generate the report.

```
Set wdApp = New Word.Application
Set wdrpt = New WDRpt
Call wdrpt.WordReport(wdApp, "c:\wordreport\myreport.wrf",
"1996-05-01", "Dairy Products")
```

WDRptCom will replace "\$SalesDate" in SQL statements with "1996-05-01", replace "\$Category" with "Dairy Products", and then submit

SQL statements to data sources.

### **6.1.3 Using a WRF file in command line**

In the WDRReportCom, there is an executable file WordReport.exe that can read a WRF file to generate a report. It is the same as WDRReportGen command line. For example, you have created the WRF file “myreport.wrf” and the template file. In the WRF file, there are two parameters. The first parameter is the sales date “\$SalesDate”, and the second is the category of the products “\$Category”. You can run WordReport.exe in command line mode as follows:

```
wordreport c:\wordreport\myreport.wrf 1996-05-01 “Dairy Products”
```

WDRReportCom will replace “\$SalesDate” in SQL statements with “1996-05-01”, replace “\$Category” with “Dairy Products”, and then submit SQL statements to data sources.

### **6.1.4 Creating a WRF file**

The WRF file is a text file. You can create and modify a WRF file in WDRReportGen or a text editor.

Sometimes you want to make a WRF file programmatically. You can write a program to create a WRF file using C, perl or DOS shell, and then run WDRReportCom to generate report. The two steps can be written into a batch file.

1. Write a program to make the WRF file as you need.
2. Write a batch file to call the program and WordReport.exe.

For example, you write a batch file runrpt.bat as follows. changewrf is an executable file that reads template.txt and output template.wrf. First runrpt.bat call changewrf to make the WRF file, and then call WordReport.exe to generate the report.

```

@echo off
if "%1"==" " goto usage
goto process
:usage
echo Usage: runrpt ReportDate
echo ReportDate   Date format 'YYYY-MM-DD'
goto :EOF
:process
changelwrf %1 <"template.txt" >"template.wrf"
WordReport "template.wrf" %1

```

### 6.1.5 Using parameters

You can use parameters in the WRF file. You can pass values to WDRReportCom when it processes a WRF file. WDRReportCom will replace the parameter names with the actual values. You can use the parameters in the SQL statements and the paths and names of the files.

To use a parameter, you must define it first. If you have defined a parameter name, you can use it in SQL statements. In fact, WDRReportCom will replace all strings that are the same as the names of the parameters. You should be careful to define a unique name for each parameter. It is a good choice a name begins with the "\$" character.

#### Example

Input an order id to get the order information. The field OrderID is numeric type.

#### 1. Defining a parameter

Define a parameter as follows:

```

Name: $OrderID
Title: Order ID (>=10248)

```

Default: 10360

## 2. Using a parameter

You can use the parameter “\$OrderID” in SQL statements. For example:

```
SELECT o.OrderID
,o.OrderDate
,SUM(d.UnitPrice * d.Quantity * (1-d.Discount)) AS Amount
FROM Orders o, OrderDetails d
WHERE o.OrderID = d.OrderID
AND o.OrderID = $OrderID
GROUP BY o.OrderID, o.OrderDate
;
```

### **Example**

Define two parameters. The first parameter is the sales date, and the second is the category of the products. The field OrderDate is date type, and CategoryName is char type.

## 1. Defining parameters

Define parameters as follows:

Name1: \$SalesDate

Title1: Sales Date

Default1: 1996-05-01

Name2: \$Category

Title2: Category of Products

Default2:

## 2. Using parameters

You can use the parameters “\$SalesDate”, “\$Category” in SQL statements.

For example:

```
SELECT .....
FROM Orders, OrderDetails, Products, Categories
```

```
WHERE .....  
AND OrderDate = '$SalesDate'  
AND CategoryName LIKE '$Category%'  
;  
/* For Microsoft Jet SQL, LIKE '$Category*' */
```

### **Example**

Get the information from the database, table and column that you identify when the report is generated.

#### 1. Defining parameters

Define parameters as follows:

Name1: \$Database

Title1: Database Name

Default1:

Name2: \$Table

Title2: Table Name

Default2:

Name3: \$Column

Title3: Column Name

Default3:

#### 2. Using parameters

You can use the parameters "\$Database", "\$Table" and "\$Column" in SQL statements. For example:

```
USE $Database;
```

or

```
DATABASE $Database;
```

```
SELECT $Column
```

```
FROM $Table
```

```
;
```

## Example

Use parameters in the path and name of the report file and the log file.

### 1. Defining a parameter

Define a parameter as follows:

Name: \$CustomerID

Title: Customer ID

Default: C000001

### 2. Using a parameter

ReportFileName=report\report\_ \$CustomerID.doc

LogFileName=log\report\_ \$CustomerID.log

or

ReportFileName=report\ \$CustomerID\report.doc

LogFileName=log\ \$CustomerID\report.log

## 6.1.6 Converting files

You can convert a file from Microsoft Word document to and from another file format. For example, the template file is a RTF file with a .rtf extension, and the report file is a HTML file with a .htm extension. Microsoft Word uses file format converters to open and save documents in different formats. The most commonly used converters are installed with Microsoft Word by default. If you want to open or save in a format that's not installed by default, you may need to install additional converters.

The file formats WDRReportCom supports can be one of these. What file format WDRReportCom supports is dependent on your Microsoft Word and converters installed. For example, Microsoft Word 2003 supports XML, but Microsoft Word 97/2000 does not support it. For more information about converting files, please refer to *Microsoft Word Help*. The file "wconv.cfg" located in the WDRReportCom directory contains the information of file formats.

You can expand it if your Microsoft Word supports more file formats.

File Format Name	Value	Description	Extension	Converter
WdFormatDocument	0	Word Document	doc	Office97
WdFormatTemplate	1	Word Template	dot	Office97
WdFormatRTF	6	Rich Text Format	rtf	Office97
WdFormatHTML	8	Web Page	htm html	Office2000
WdFormatFilteredHTML	10	Web Page Filtered	htm html	Office2003
WdFormatWebArchive	9	Single File Web Page	mht mhtml	Office2003
WdFormatXML	11	XML Document	xml	Office2003
WdFormatText	2	Text	txt	Office97
WdFormatTextLineBreaks	3	Text (Line Breaks)	txt	Office97
WdFormatDOSText	4	Text (DOS)	txt	Office97
WdFormatDOSTextLineBreaks	5	Text (DOS Line Breaks)	txt	Office97
WdFormatUnicodeText	7	Unicode Text	txt	Office97
MSWordWin2		Word 2.x for Windows	doc	External
MSWordWin5		Word 5.0 for Windows	doc	External
MSWord6RTFExp		Word 97-2003 & 6.0/95 – RTF	doc	External
wks632		Works 6.0 & 7.0	wps	External
MSWorksWin6		Works 6.0 & 7.0	wtf	External
Works2001		Works 7.0	wps	External
WrdPrfctDos		WordPerfect 5.x	doc	External
WordPerfect6x		WordPerfect 6.x	wpd doc	External

For Microsoft Word 2007, please copy “wconv2007.cfg” to “wconv.cfg”. This file contains the information of file formats for Microsoft Word 2007.

File Format Name	Value	Description	Extension
WdFormatXMLDocument	12	Word Document	Docx
WdFormatXMLDocumentMacroEnabled	13	Word Macro-enabled Document	Docm
WdFormatDocument	0	Word 97-2003 Document	Doc
WdFormatXMLTemplate	14	Word Template	Dotx
WdFormatXMLTemplateMacroEnabled	15	Word Macro-enabled Document Template	Dotm
WdFormatTemplate	1	Word 97-2003 Template	Dot
WdFormatRTF	6	Rich Text Format	Rtf
WdFormatHTML	8	Web Page	htm html
WdFormatFilteredHTML	10	Web Page Filtered	htm html
WdFormatWebArchive	9	Single File Web Page	mht mhtml
WdFormatPDF	17	PDF	Pdf
WdFormatXPS	18	XPS Document	Xps
WdFormatFlatXML	19	Word XML Document	Xml
WdFormatFlatXMLMacroEnabled	20	Word XML Macro-enabled	Xml

		Document	
WdFormatXML	11	Word 2003 XML Document	Xml
WdFormatText	2	Text	Txt
WdFormatTextLineBreaks	3	Text (Line Breaks)	Txt
WdFormatDOSText	4	Text (DOS)	Txt
WdFormatDOSTextLineBreaks	5	Text (DOS Line Breaks)	Txt
WdFormatUnicodeText	7	Unicode Text	Txt
MSWorksWin6		Works 6.0 - 9.0	Wtf
wks632		Works 6.0 - 9.0	Wps
Works2001		Works 7.0	Wps
WrdPrfctDos		WordPerfect 5.x	Doc
WordPerfect6x		WordPerfect 6.x	wpd doc

Note: For an external converter, the file format name is the class name of the file converter, and the format value may not be the same on another computer. So do not assign a number to the format value.

## 6.2 WRF File Reference

### 6.2.1 WRF File Format

The layout of a WRF file is as the following:

WordReport Version 2.0

[Data Source]

.....

[File]

.....

[Parameter]

.....

[SQL]

.....

“WordReport” is the flag of the WRF file. “Version 2.0” is the version of the WRF file.

A WRF file contains several sections. The sections of [Data Source], [File],

and [Parameter] consist of a group of related settings. The sections and settings are listed in the WRF file in the following format:

```
[section name]
  keyname=value
```

In this example, [section name] is the name of a section. The enclosing brackets ([]) are required, and the left bracket must be in the leftmost column on the screen.

The keyname=value statement defines the value of each setting. A keyname is the name of a setting. It can consist of any combination of letters and digits, and must be followed immediately by an equal sign (=). The value can be an integer, a string, or a quoted string, depending on the setting.

You can include comments in these sections. You must begin each line of a comment with a semicolon (;).

The [SQL] section consists of functions. Each function is begin with the "@" character. Syntax:

```
@functionno=functionname(arguments)
  sqlstatement
```

The *functionno* is the label of the function.

The *functionname* represents a function.

The *arguments* define various properties for the function. An argument takes the form *Name="Value"*. The argument value can be delimited by single or double quotes.

The *sqlstatement* is a SQL statement.

You can use comments in [SQL] section. A comment is the "/\*" characters, followed by any sequence of characters (including new lines), followed by the "\*" characters. You cannot nest comments.

## 6.2.2 [Data Source] Section

The [Data Source] section contains information how to connect to data sources.

*Name1=<name1>*

*Name2=<name2>*

.....

*Name10=<name10>*

These settings specify the names of data sources you want to connect to. Name1 specifies the name of the first data source. Name2 specifies the name of the second data source..... You can define up to 10 data sources in one WRF file. You can make a connection to a data source using an ODBC data source name or a connection string. Even if you use a connection string to make a connection, you should define a name that you can reference in functions.

*User1=<username1>*

*User2=<username2>*

.....

*User10=<username10>*

These settings specify the user names. If you use an ODBC data source name to make a connection, you should define user name and password. If you use a connection string to make a connection, WDRReportCom will ignore the setting. User1 specifies the user name of the first data source. User2 specifies the user name of the second data source..... They are optional settings. If defined default user and password in ODBC data source, you may not define them.

*Password1=<password1>*

*Password2=<password2>*

.....

*Password10=<password10>*

These settings specify the user passwords. If you use an ODBC data source name to make a connection, you should define user name and password. If you use a connection string to make a connection, WDRReportCom will ignore the setting. Password1 specifies the password of the first data source.

Password2 specifies the password of the second data source..... They are optional settings. If defined default user and password in ODBC data source, you may not define them.

*ConnectionString1=<connectionstring1>*

*ConnectionString2=<connectionstring2>*

.....

*ConnectionString10=<connectionstring10>*

These settings specify the connection strings. If you defined a connection string, WDRReportCom will make a connection to the data source using the connection string, and ignore the settings of the name, user and password. But you must define a name that you can reference in functions.

ConnectionString1 specifies the connection string of the first data source.

ConnectionString2 specifies the connection string of the second data source..... They are optional settings. If no connection string, WDRReportCom will make a connection to data source using the ODBC data source name.

*EncryptPassword =Y/N*

This setting specifies how to save the passwords of the data sources. If the

value is Y, passwords will be saved in an encrypted format. If the value is N, the passwords will be saved in plain text.

### **6.2.3 [FILE] Section**

[FILE] section contains information about files.

*ReportTemplateName=<templatefilename>*

This setting specifies the name of the report template file. <templatefilename> value is the name and path of the report template file. The file path can be a relative path or an absolute path. If it is a relative path, the base path is the path of the WRF file.

*ReportFileName=<reportfilename>*

This setting specifies the name of the report file. <reportfilename> value is the name and path of the report file. The file path can be a relative path or an absolute path. If it is a relative path, the base path is the path of the WRF file. In <reportfilename>, you can use parameters.

*ReportFileType=<reportfiletype>*

This setting specifies the type of the report file. <reportfiletype> value is the name or value of the file format. For example, wdFormatRTF or 6. What file format WDRReportCom supports is dependent on your Microsoft Word and converters installed.

*ProtectReport=Y/N*

This setting specifies whether the report generated is protected. If the value is Y, the report is protected, and can not be modified. If the value is N, the report is not protected. Default is N.

*ProtectionPassword=<protectionpassword>*

This setting specifies the password that is used to protect the report. <protectionpassword> value is the password. This setting is valid when ProtectReport is Y. If there is not this setting and ProtectReport is Y, a random password will be created.

*LogFileName=<logfilename>*

This setting specifies the name of the log file. <logfilename> value is the name and path of the log file. The file path can be a relative path or an absolute path. If it is a relative path, the base path is the path of the WRF file. In <logfilename>, you can use parameters.

## **6.2.4 [PARAMETER] Section**

[PARAMETER] section contains information about parameters.

*Name1=<name1>*

*Name2=<name2>*

.....

*Name10=<name10>*

These settings specify the names of the parameters. Name1 specify the name of the first parameter, Name2 specifies the name of the second parameter.....

You can define up to 10 parameters in one WRF file.

*Title1=<title1>*

*Title2=<title2>*

.....

*Title10=<title10>*

These settings specify the titles of the parameters. Title1 specifies the title of the first parameter. Title2 specifies the title of the second parameter.....

*Default1=<default1>*

*Default2=<default2>*

.....

*Default10=<default10>*

These settings specify the default values of the parameters. Default1 specifies the default value of the first parameter. Default2 specifies the default value of the second parameter.....

## **6.3 Function Reference**

### **6.3.1 Fixed Table Report**

Uses FixTableReport method to generate a fixed table report. In a fixed table report, the number of rows and columns is fixed. WDRReportCom executes a SQL statement to get data from data source, and directly fills data into the cells of a table.

#### **Syntax**

*Report(...)*

*sqlstatement*

#### **Arguments**

TYPE = "fix"

TABLE = *table*

FILLORDER = *fillorder*

CELL= *celllist*

RANGE = *range*

IMAGE = *fieldlist*

PAGEBREAK = *pagelength*

CONNECT = *datasource*

The **TYPE** argument specifies the report type. "fix" means a fixed table report. The **TABLE** argument identifies a table in the report template. The *table* is the index number of the table or the bookmark name in the table. The index number starts at 1. For examples, table 2 is the second table in the document. The index number of a nested table likes 2-1-2. For examples, table 2-1 is the first table inside table 2, and table 2-1-2 is the second table inside table 2-1. The max nested level WDRReportCom supports is 3.

The **FILLORDER** argument specifies the order in which WDRReportCom fill data. Possible values are row or col. "row" means to fill data by rows, and "col" means to fill data by columns. Default is row.

The **CELL** argument specifies the positions where data values will be inserted. The *celllist* is the list of cells separated by the “,” character. For example, “A2,B2,B3,D2,D3”. The cells in the *celllist* should correspond to the data source fields in the SQL statement. The value of the first field is put into the first cell, and the value of the second field is put into the second cell ..... WDRReportCom will use the next cell if you omit a cell except the first cell. If FILLORDER=“row”, the next cell is the right cell. If FILLORDER=“col”, the next cell is the below cell.

The **RANGE** argument specifies the range in the table to be used for the records. WDRReportCom will skip the range for each record. You can reference a range of cells like “2:4” or “B:D”. The default range is the area that includes all cells for the records.

The **IMAGE** argument specifies the data source fields are picture files. The *fieldlist* is the list of data source fields separated by the “,” character. You can identify a field using the name of field or the index number of field, but not

simultaneously. In data source, you stored the path and file name of the picture, not the picture. The file path can be a relative path, an absolute path or a URL. If it is a relative path, the base path is the path of the report template file.

The **PAGEBREAK** argument specifies the page breaks. The unit of page length is r that means record. For example, “6r” or “6” means that WDRReportCom will insert a page break per 6 records. Default is no page break.

The **CONNECT** argument specifies the connection to a data source. The CONNECT can takes a string that expresses a data source name or a number that expresses a data source index. The index number of data source is the sequential number defined in the WRF file, and starts at 1. The default implies the first data source.

The **sqlstatement** is a SQL statement such as a SELECT statement.

### Example

This example uses Fixed Table Report function to make the report “Top 5 Employees for Sales”.

```
@F1=REPORT(type=fix table=6 cell=B2)
```

```
SELECT TOP 5 e.FirstName + ' ' + e.LastName
```

```
    , SUM(d.Quantity)
```

```
    , Sum(d.UnitPrice * d.Quantity * (1-d.Discount)) AS SalesAmount
```

```
FROM Orders o
```

```
    ,OrderDetails d
```

```
    ,Products p
```

```
    ,Employees e
```

```
WHERE o.OrderID = d.OrderID
```

```
AND d.ProductID = p.ProductID
```

```
AND o.EmployeeID = e.EmployeeID
AND YEAR(o.OrderDate) = 1996
AND MONTH(o.OrderDate) = 04
GROUP BY e.FirstName, e.LastName
ORDER BY 3 DESC
```

### **6.3.2 Variable Table Report**

Uses VarTableReport method to generate a variable table report. In a variable table report, the number of rows or columns in the table is unfixed, and it is variable as the number of the result records. WDRReportCom executes a SQL statement to get data from data source, inserts some blank rows/columns or copy a range for each record, then fills data into the cells of a table.

#### **Syntax**

```
Report(...)  
sqlstatement
```

#### **Arguments**

```
TYPE = "var"  
TABLE = table  
FILLORDER = fillorder  
CELL= celllist  
RANGE = range  
IMAGE = fieldlist  
RESERVE = reserverecords  
PAGEBREAK = pagelength  
NODATA = nodataoption  
CONNECT = datasource
```

The **TYPE** argument specifies the report type. "var" means a variable table report. Default is var.

The **TABLE** argument identifies a table in the report template. The *table* is the index number of the table or the bookmark name in the table. The index number starts at 1. For examples, table 2 is the second table in the document. The index number of a nested table likes 2-1-2. For examples, table 2-1 is the first table inside table 2, and table 2-1-2 is the second table inside table 2-1. The max nested level WDRReportCom supports is 3.

The **FILLORDER** argument specifies the order in which WDRReportCom fill data. Possible values are row or col. "row" means to fill data by rows, and "col" means to fill data by columns. Default is row.

The **CELL** argument specifies the positions where data values will be inserted. The *celllist* is the list of cells separated by the “,” character. For example, “A2,B2,B3,D2,D3”. The cells in the *celllist* should correspond to the data source fields in the SQL statement. The value of the first field is put into the first cell, and the value of the second field is put into the second cell ..... WDRReportCom will use the next cell if you omit a cell except the first cell. If FILLORDER=“row”, the next cell is the right cell. If FILLORDER=“col”, the next cell is the below cell.

The **RANGE** or **COPYRANGE** argument specifies the range in the table to be used for the records. WDRReportCom will skip or repeat the range for each record. You can reference a range of cells like “2:4” or “B:D”. The default range is the area that includes all cells for the records. For Range argument, WDRReportCom will insert the blank rows/columns of the range for each record. For COPYRANGE argument, it will copy the original range and insert the copied range for each record.

The **IMAGE** argument specifies the fields are picture files. The *fieldlist* is the list of data source fields separated by the “,” character. You can identify a field

using the name of field or the index number of field, but not simultaneously. In data source, you stored the path and file name of the picture, not the picture.

The file path can be a relative path, an absolute path or a URL. If it is a relative path, the base path is the path of the report template file.

The **RESERVE** argument specifies the number of the records for which you reserved some rows/columns in the report template for the report. The *reserverecords* represents the number of the records you reserved in the report template. Possible values are 1 or 2. One means you reserved some rows/columns for one record, and two means some rows/columns for two records. Default is 1.

The **PAGEBREAK** argument specifies the page breaks. The unit of page length is r that means record. For example, "6r" or "6" means that WDRReportCom will insert a page break per 6 records. Default is no page break.

The **NODATA** argument specifies an option when no data are returned from data source. If the value is "delrange", WDRReportCom will delete the range when no data are returned. "deltable" means to delete the table. Default is to do nothing.

The **CONNECT** argument specifies the connection to a data source. The **CONNECT** can takes a string that expresses a data source name or a number that expresses a data source index. The index number of data source is the sequential number defined in the WRF file, and starts at 1. The default implies the first data source.

The ***sqlstatement*** is a SQL statement such as a SELECT statement.

### **Example**

This example uses Variable Table Report function to make the report "Mail Label".

```

@F1=Report(type=var table=1 cell=B7,B8,B9,B10 copyrange=1:11
pagebreak = 4r)
SELECT CompanyName
,Address
,CityName & ', ' & CountryName
,PostalCode
FROM Customers, Cities, Countries
WHERE Customers.CityCode = Cities.CityCode
AND Customers.CountryCode = Cities.CountryCode
AND Customers.CountryCode = Countries.CountryCode
ORDER BY CompanyName

```

### 6.3.3 Group Table Report

Uses GroupTableReport method to generate a variable table report and group data. In a variable table report, the number of rows or columns in the table is unfixed, and it is variable as the number of the result records. WDRReportCom executes a SQL statement to get data from data source, copy the group range for each group, copy the detail range for each record, then fills data into the table.

#### Syntax

```

Report(...)
sqlstatement

```

#### Arguments

```

TYPE = "var"
TABLE = table
FILLORDER = fillorder
CELL= celllist
RANGE = range

```

GROUP= *grouplist*

GROUPRANGE = *grouprange*

IMAGE = *fieldlist*

PAGEBREAK = *pagelength*

NODATA = *nodataoption*

CONNECT = *datasource*

The **TYPE** argument specifies the report type. "var" means a variable table report. Default is var.

The **TABLE** argument identifies a table in the report template. The *table* is the index number of the table or the bookmark name in the table. The index number starts at 1. For examples, table 2 is the second table in the document.

The index number of a nested table likes 2-1-2. For examples, table 2-1 is the first table inside table 2, and table 2-1-2 is the second table inside table 2-1.

The max nested level WDRReportCom supports is 3.

The **FILLORDER** argument specifies the order in which WDRReportCom fill data. Possible values are row or col. "row" means to fill data by rows, and "col" means to fill data by columns. Default is row.

The **CELL** argument specifies the positions where data values will be inserted.

The *celllist* is the list of cells separated by the “,” character. For example, “A2,B2,B3,D2,D3”. The cells in the *celllist* should correspond to the data source fields in the SQL statement. The value of the first field is put into the first cell, and the value of the second field is put into the second cell .....

WDRReportCom will use the next cell if you omit a cell except the first cell. If FILLORDER=“row”, the next cell is the right cell. If FILLORDER=“col”, the next cell is the below cell.

The **RANGE** or **COPYRANGE** argument specifies the range in the table to be used for the details. WDRReportCom will skip or repeat the range for each record. You can reference a range of cells like “2:4” or “B:D”. The default

range is the area that includes all cells for the details. For Range argument, WDRReportCom will insert the blank rows/columns of the range for each record. For COPYRANGE argument, it will copy the original range and insert the copied range for each record. But if the range of any group is not same as the range of the details, RANGE is same as COPYRANGE.

The **GROUP** argument specifies the group of the report. The *grouplist* is the list of data source fields separated by the “,” character. You can identify a field using the name or index number of the field, but not simultaneously. In one report, there may be up to 10 groups. Notes: the order of the groups should be in accordance with the order of the ORDER BY clause in the SQL statement.

The **GROUPRANGE** argument follows the GROUP argument, and specifies the range of the group in the table. For example, the grouprange of level 1 must follow the group of level 1, and the grouprange of level 2 must follow the group of level 2. WDRReportCom will repeat the group range for each group. The range of the group should contain the range of the details and the area that includes all cells for this group. You reference a group range like “2:4” or “B:D”. For example, there are two groups, the range of the group one contains all cells for the group one and the range of the group two, and the range of the group two contains all cells for the group two and the range of the details. The default range is the area that includes all cells for this group and the range or group range for the lower level group.

The **IMAGE** argument specifies the fields are picture files. The *fieldlist* is the list of data source fields separated by the “,” character. You can identify a field using the name of field or the index number of field, but not simultaneously. In data source, you stored the path and file name of the picture, not the picture. The file path can be a relative path, an absolute path or a URL. If it is a relative path, the base path is the path of the report template file.

The **PAGEBREAK** argument specifies the page breaks. The unit of page length is r or g. "r" means record, "g1" means group one, "g2" means group two..... For example, "6r" or "6" means that WDRReportCom will insert a page break per 6 records, "1g1" or "1g" means a page break per group one, and "1g1,6r" means a page break per group one or 6 records. Default is "" that means no page break.

The **NODATA** argument specifies an option when no data are returned from data source. If the value is "delrange", WDRReportCom will delete the range when no data are returned. "deltable" means to delete the table. Default is to do nothing.

The **CONNECT** argument specifies the connection to a data source. The CONNECT can takes a string that expresses a data source name or a number that expresses a data source index. The index number of data source is the sequential number defined in the WRF file, and starts at 1. The default implies the first data source.

The **sqlstatement** is a SQL statement such as a SELECT statement.

### **Example**

This example uses Group Table Report function to make the report "Customer Profile".

```
@F1= Report(table=1 cell=A2,B3,C3,D3,D4,E3,E4,E5
copyrange=2:5 group=1 pagebreak = 5r)
SELECT LEFT(CompanyName,1)
,CompanyName
,ContactName
,'Phone: ' & Phone
,'Fax: ' & Fax
,Address
```

```
,CityName & ', ' & CountryName  
,PostalCode  
FROM Customers, Cities, Countries  
WHERE Customers.CityCode = Cities.CityCode  
AND Customers.CountryCode = Cities.CountryCode  
AND Customers.CountryCode = Countries.CountryCode  
ORDER BY CompanyName
```

### 6.3.4 Form Report

Uses FormReport method to generate a form report and group data. For a form report, you can put data from data source as text, list, title and table in the report file. WDRReportCom gets data from a recordset object, copy the group range for each group, and copy the detail range for each record.

#### Syntax

```
Report(...)  
sqlstatement
```

#### Arguments

```
TYPE = "form"  
CELL= celllist  
RANGE = range  
GROUP= grouplist  
GROUPRANGE = grouprange  
IMAGE = fieldlist  
PAGEBREAK = pagelength  
NODATA = nodataoption  
CONNECT = datasource
```

The **TYPE** argument specifies the report type. "form" means a form report.

The **CELL** argument specifies the positions where data values will be inserted.

The *celllist* is the list of merge fields or quote fields separated by the “,” character. For example, “ProductName,ProductID,QuantityPerUnit,UnitPrice”. The merge fields or quote fields in the *celllist* should correspond to the data source fields in the SQL statement. The value of the first data source field is put into the first merge field or quote field, and the value of the second data source field is put into the second merge field or quote field .....

The **RANGE** argument specifies the range to be used for the records.

WDRReportCom will repeat the range for each record. A range is defined by a bookmark. You reference a range using a bookmark name. The default range is the group range or the entire document.

The **GROUP** argument specifies the group of the report. The *grouplist* is the list of data source fields separated by the “,” character. You can identify a field using the name of field or the index number of field, but not simultaneously. In one report, there may be up to 10 groups. The first GROUP is group one, the second is group two..... Notes: the order of groups should be in accordance with the order of ORDER BY clause in the SQL statement.

The **GROUPRANGE** argument follows the GROUP argument, and specifies the range of the group in the table. For example, the grouprange of level 1 must follow the group of level 1, and the grouprange of level 2 must follow the group of level 2. WDRReportCom will repeat the range for each group. A range is defined by a bookmark. You reference a range using a bookmark name.

The range of the group should contain the range of the details and the area that includes all merge fields or quote fields for this group. For example, there are two groups, the range of the group one contains all merge fields or quote fields for the group one and the range of the group two, and the range of the group two contains all merge fields or quote fields for the group two and the range of the details. The default range is the range of the upper level group or the entire document.

The **IMAGE** argument specifies the fields are picture files. The *fieldlist* is the list of data source fields separated by the “,” character. You can identify a field using the name of field or the index number of field, but not simultaneously. In data source, you stored the path and file name of the picture, not the picture. The file path can be a relative path, an absolute path or a URL. If it is a relative path, the base path is the path of the report template file.

The **PAGEBREAK** argument specifies the page breaks. The unit of page length is r or g. "r" means record, "g1" means group one, "g2" means group two..... For example, “6r” or “6” means that WDRReportCom will insert a page break per 6 records, “1g1” or “1g” means a page break per group one, and “1g1,6r” means a page break per group one or 6 records. Default is “” that means no page break.

The **NODATA** argument specifies an option when no data are returned from data source. If the value is "delrange", WDRReportCom will delete the range when no data are returned. Default is to do nothing.

The **CONNECT** argument specifies the connection to a data source. The CONNECT can takes a string that expresses a data source name or a number that expresses a data source index. The index number of data source is the sequential number defined in the WRF file, and starts at 1. The default implies the first data source.

The **sqlstatement** is a SQL statement such as a SELECT statement.

### **Example**

This example uses Form Report function to make the report “Product Catalog”.

```
@F1=Report(type=form cell=CategoryName,Description
,ProductName,ProductID,QuantityPerUnit,UnitPrice
range=Product group=1,2 grouprange=Category)
```

```
SELECT CategoryName
,Description
,ProductName
,ProductID
,QuantityPerUnit
,UnitPrice
FROM Products, Categories
WHERE Products.CategoryID = Categories.CategoryID
ORDER BY 1,3
```

### 6.3.5 MSGraph Chart

Uses MSGraphChart method to generate a chart. WDRReportCom gets data from a recordset object, and fills data into the datasheet of a chart in the report file.

#### Syntax

```
Chart(...)  
sqlstatement
```

#### Arguments

```
CHART = chart  
FILLORDER = fillorder  
CELL= celllist  
RANGE = range  
CONNECT = datasource
```

The **CHART** argument identifies a chart in the report template. The *chart* is the index number or the bookmark name of the chart. The index number starts at 1. For examples, chart 2 is the second chart in the document. You can reference a chart by a bookmark. For examples, chart="Chart1". "Chart1" is the bookmark of a chart.

The **FILLORDER** argument specifies the order in which WDRReportCom fill data. Possible values are row or col. "row" means to fill data by rows, and "col" means to fill data by columns. Default is col.

The **CELL** argument specifies the positions where data values will be inserted.

The *celllist* is the list of cells separated by the “,” character. For example, “A2,B2,B3,D2,D3”. The cells in the *celllist* should correspond to the data source fields in the SQL statement. The value of the first field is put into the first cell, and the value of the second field is put into the second cell .....

WDRReportCom will use the next cell if you omit a cell except the first cell. If FILLORDER=“row”, the next cell is the right cell. If FILLORDER=“col”, the next cell is the below cell. Note: On the datasheet, the leftmost column and the top row, which are commonly used for legend text or axis labels, are referred to as column 0 (zero) and row 0 (zero).

The **RANGE** argument specifies the range in the datasheet of the chart to be used for the records. WDRReportCom will skip the range for each record. You can reference a range of cells like “2:4” or “B:D”. The default range is the area that includes all cells for the records.

The **CONNECT** argument specifies the connection to a data source. The CONNECT can takes a string that expresses a data source name or a number that expresses a data source index. The index number of data source is the sequential number defined in the WRF file, and starts at 1. The default implies the first data source.

The *sqlstatement* is a SQL statement such as a SELECT statement.

### **Example**

This example uses Chart function to make the chart “Sales by Categories”.

```
@F3_2=CHART(chart=Chart3 cell=A0)
```

```
SELECT c.CategoryName  
      , Sum(d.UnitPrice * d.Quantity * (1-d.Discount))
```

```
FROM Orders o
    ,OrderDetails d
    ,Products p
    ,Categories c
WHERE o.OrderID = d.OrderID
AND d.ProductID = p.ProductID
AND p.CategoryID = c.CategoryID
AND YEAR(o.OrderDate) = 1996
AND MONTH(o.OrderDate) = 04
GROUP BY c.CategoryName
ORDER BY c.CategoryName
```

### 6.3.6 Excel Chart

Uses ExcelChart method to generate a chart. WDRReportCom gets data from a recordset object, and fills data into the worksheet of a chart in the report file.

#### Syntax

```
Chart(...)
sqlstatement
```

#### Arguments

```
CHART = chart
TYPE = type
FILLORDER = fillorder
CELL= celllist
RANGE = range
CONNECT = datasource
```

The **CHART** argument identifies a chart in the report template. The *chart* is the index number or the bookmark name of the chart. The index number starts at 1. For examples, chart 2 is the second chart in the document. You can

reference a chart by a bookmark. For examples, chart="Chart1". "Chart1" is the bookmark of a chart.

The **TYPE** argument specifies the report type. Possible values are fix or var. "fix" means that WDRReportCom will directly fill data values into the worksheet of the chart. "var" means that WDRReportCom will add some blank rows/columns before filling data values into the worksheet of the chart. Default is var. When the report type is "var", you should reserve two rows/columns in the worksheet in the report template, and set the data range of the chart to 2 rows/columns. The RESERVE must be 2.

The **FILLORDER** argument specifies the order in which WDRReportCom fill data. Possible values are row or col. "row" means to fill data by rows, and "col" means to fill data by columns. Default is row.

The **CELL** argument specifies the positions where data values will be inserted. The *celllist* is the list of cells or fields separated by the "," character. The *celllist* identifies the cells in a worksheet. For example, "A2,B2,B3,D2,D3". The cells in the *celllist* should correspond to the data source fields in the SQL statement. The value of the first field is put into the first cell, and the value of the second field is put into the second cell ..... WDRReportCom will use the next cell if you omit a cell except the first cell. If FILLORDER="row", the next cell is the right cell. If FILLORDER="col", the next cell is the below cell.

The **RANGE** argument specifies the range in the worksheet of the chart to be used for the records. WDRReportCom will skip the rows/columns of the range for each record. A range is composed of some rows or columns. You can reference a range of cells like "2:4" or "B:D". The default range is the area that includes all cells for details. For a variable table report, WDRReportCom will insert the blank rows/columns of the range for each record.

The **CONNECT** argument specifies the connection to a data source. The CONNECT can takes a string that expresses a data source name or a number

that expresses a data source index. The index number of data source is the sequential number defined in the WRF file, and starts at 1. The default implies the first data source.

The ***sqlstatement*** is a SQL statement such as a SELECT statement.

### **Example**

The following function makes the chart: Sales by Categories.

```
@F3_2=CHART(chart=Chart3 cell=A2)
SELECT c.CategoryName
       , Sum(d.UnitPrice * d.Quantity * (1-d.Discount))
FROM Orders o
       ,OrderDetails d
       ,Products p
       ,Categories c
WHERE o.OrderID = d.OrderID
AND d.ProductID = p.ProductID
AND p.CategoryID = c.CategoryID
AND YEAR(o.OrderDate) = 1996
AND MONTH(o.OrderDate) = 04
GROUP BY c.CategoryName
ORDER BY c.CategoryName
```

### **6.3.7 DocVariable**

Uses SetDocVariable method to assigns the values to the docvariables defined in the Word document. WDRReportCom will just fetch the first record, no matter how many records are returned from data source.

#### **Syntax**

```
DocVariable(...)
sqlstatement
```

## Arguments

NAME= *namelist*

CONNECT= *datasource*

The **NAME** argument specifies the name of the document variables you want assign values to. The *namelist* is the list of variable names separated by the “,” character. For example, “BeginDate, EndDate” means two document variables: BeginDate and EndDate that should be defined in the report template. The variables in the *namelist* should correspond to the fields in the SQL statement. The value of the first field is put into the first variable, and the value of the second field is put into the second variable ...

The **CONNECT** argument specifies the connection to a data source. The CONNECT can takes a string that expresses a data source name or a number that expresses a data source index. The index number of data source is the sequential number defined in the WRF file, and starts at 1. The default implies the first data source.

The ***sqlstatement*** is a SQL statement such as a SELECT statement.

## Remarks

DocVariable function supports headers and footers. You can use it to put data into headers or footers.

## Example

This example uses DocVariable function to assign the values of min\_date and max\_date to the document variables: BeginDate and EndDate.

```
@F1=DOCVARIABLE(NAME=BeginDate,EndDate)
```

```
SELECT min_date, max_date
```

```
FROM tmp0
```

### 6.3.8 ExecSQL

Executes a SQL statement, but no data is returned to the report.

#### Syntax

```
ExecSQL(...)  
sqlstatement
```

#### Arguments

```
CONNECT= datasource
```

The **CONNECT** argument specifies the connection to a data source. The **CONNECT** can takes a string that expresses a data source name or a number that expresses a data source index. The index number of data source is the sequential number defined in the WRF file, and starts at 1. The default implies the first data source.

The ***sqlstatement*** is a SQL statement that can be DDL (Data Definition Language), DML (Data Manipulation Language) and even DCL (Data Control Language).

Using EXECSQL function, you can open a database, create a temporary table, insert data into a temporary table, update data, execute a stored procedure, and drop a table. It is very useful to create a temporary table, and prepare data for REPORT function.

#### Example

This example uses ExecSQL functions to create a table tmp0, and add some records into the table. No result is returned to the report.

```
@F1=EXECSQL()  
CREATE TABLE tmp0 (  
min_date DATE,  
max_date DATE)  
;
```

```
@F2=EXECSQL()  
INSERT INTO tmp0  
SELECT ...
```

# Chapter 7 Advanced Reports

## 7.1 Executing multiple SQL statements

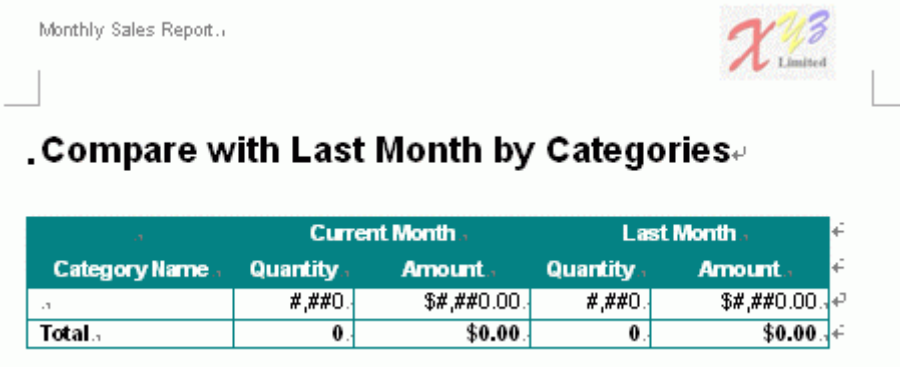
In one report building process, WDRReportCom can execute multiple SQL statements. This enables you to

1. Create a report like building block. You may divide one report into several parts, and respectively use the different SQL statements to make each part of the report. You can use the different queries to get the data located in the different tables or databases.
2. Create a complex report using the temporary table. First, you create a temporary table. Second, use several SQL statements to prepare data in the temporary table. You can execute INSERT, UPDATE, DELETE, INSERT SELECT statements. And then put the prepared data from the temporary table into your report.
3. Create one report file with several reports. For example, you may create one document with several tables.


### Example

This example executes multiple SQL statements to create a report.

1. Create the template in Microsoft Word.



Monthly Sales Report .i



**. Compare with Last Month by Categories**

Category Name	Current Month		Last Month	
	Quantity	Amount	Quantity	Amount
	###0.	\$#,##0.00.	###0.	\$#,##0.00.
<b>Total</b>	<b>0</b>	<b>\$0.00</b>	<b>0</b>	<b>\$0.00</b>

2. Write SQL statements in a WRF file.

```
/******
```

Compare with Last Month by Categories

```
*****/
```

```
@F9_1=EXECSQL()
```

```
DROP TABLE tmp_category_sales
```

```
/* Create table tmp_category_sales */
```

```
@F9_2=EXECSQL()
```

```
CREATE TABLE tmp_category_sales (
```

```
CategoryID INTEGER,
```

```
Quantity INTEGER,
```

```
Amount MONEY
```

```
)
```

```
/* Get the sales amount by categories in the current month */
```

```
@F9_3=EXECSQL()
```

```
INSERT INTO tmp_category_sales (CategoryID, Quantity, Amount)
```

```
SELECT p.CategoryID
```

```
    , SUM(d.Quantity)
```

```
    , Sum(d.UnitPrice * d.Quantity * (1-d.Discount))
```

```
FROM Orders o
```

```
    ,OrderDetails d
```

```
    ,Products p
```

```
WHERE o.OrderID = d.OrderID
```

```
AND d.ProductID = p.ProductID
```

```
AND YEAR(o.OrderDate) = YEAR('1996-04-01')
```

AND MONTH(o.OrderDate) = MONTH('1996-04-01')

GROUP BY p.CategoryID

*/\* Show the sales amount by categories in the current month \*/*

@F9\_4=REPORT(table=Report9 type=var cell=A3 copyrange=3:3)

SELECT c.CategoryName

, IIF(IsNull(t.Quantity),0,t.Quantity)

, IIF(IsNull(t.Amount),0,t.Amount)

FROM Categories c LEFT JOIN tmp\_category\_sales t

ON c.CategoryID = t.CategoryID

ORDER BY c.CategoryName

*/\* Delete from table tmp\_category\_sales \*/*

@F9\_5=EXECSQL()

DELETE FROM tmp\_category\_sales

*/\* Get the sales amount by categories in the last month \*/*

@F9\_6=EXECSQL()

INSERT INTO tmp\_category\_sales (CategoryID, Quantity, Amount)

SELECT p.CategoryID

, SUM(d.Quantity)

, Sum(d.UnitPrice \* d.Quantity \* (1-d.Discount))

FROM Orders o

,OrderDetails d

,Products p

WHERE o.OrderID = d.OrderID

AND d.ProductID = p.ProductID

AND o.OrderDate >= DateAdd('m',-1,#1996-04-01#)

AND o.OrderDate < #1996-04-01#

GROUP BY p.CategoryID

/\* Show the sales amount by categories in the last month \*/

@F9\_7=REPORT(table=Report9 type=fix cell=D3)

SELECT IIF(IsNull(t.Quantity),0,t.Quantity)

, IIF(IsNull(t.Amount),0,t.Amount)

FROM Categories c LEFT JOIN tmp\_category\_sales t

ON c.CategoryID = t.CategoryID

ORDER BY c.CategoryName

3. Generate the report.

Monthly Sales Report



**. Compare with Last Month by Categories**

Category Name	Current Month		Last Month	
	Quantity	Amount	Quantity	Amount
Beverages	925	\$27,761.58	834	\$34,599.15
Condiments	378	\$10,773.27	289	\$6,293.97
Confections	880	\$22,877.18	475	\$10,074.09
Dairy Products	581	\$13,685.33	506	\$11,454.50
Grains/Cereals	189	\$3,325.40	219	\$4,060.01
Meat/Poultry	92	\$4,083.66	344	\$23,334.05
Produce	351	\$13,031.20	37	\$1,172.80
Seafood	669	\$9,316.55	584	\$12,531.12
<b>Total</b>	<b>4,065</b>	<b>\$104,854.47</b>	<b>3,288</b>	<b>\$103,519.69</b>

## 7.2 Using Formula Fields

In Microsoft Word, you can use (Formula) field to calculates a number. To insert an (Formula) field in a table or in regular text, you can use the **Formula** command (**Table** menu) or press CTRL+F9. For example, the following formula multiplies 2 by 3 and then adds 5 to the result.

{ = 5+2\*3 \# "#,##0.00"}

In an (Formula) field, an expression that can contain any combination of numbers, bookmarks that refer to numbers, fields resulting in numbers, and the available operators and functions. The expression can refer to values in a table and values returned by functions. For more detail information about (Formula) field, refer to *Microsoft Word Help*.

In a report template file, you can use all kind of (Formula) field. And they will be brought to the final report file.

### Example

Add totals such as Total Quantity, Total Amunt.

You can use (Formula) field of Microsoft Word.

1. Create a template file as follows, define the formula of total quantity as "=SUM(ABOVE)" in cell B3, and the formula of total amount as "=SUM(ABOVE)" in cell C3.

Category Name	Quantity	Amount
	#,##0	\$#,##0.00
<b>Total</b>	{ =SUM(ABOVE) \#	{ =SUM(ABOVE) \#

2. Write report function as follow.

```
@F2=REPORT(table=Report2 type=var cell=A2)
```

```
SELECT c.CategoryName, SUM(d.Quantity), Sum(d.UnitPrice * d.Quantity *  
(1-d.Discount))
```

```
FROM Orders o
```

```
    ,OrderDetails d
```

```
    ,Products p
```

```
    ,Categories c
```

```
WHERE o.OrderID = d.OrderID
```

```
AND d.ProductID = p.ProductID
```

```
AND p.CategoryID = c.CategoryID
```

```
AND YEAR(o.OrderDate) = YEAR('1996-04-01')
```

AND MONTH(o.OrderDate) = MONTH('1996-04-01')

GROUP BY c.CategoryName

ORDER BY c.CategoryName

3. The report generated by WDRReportCom is as follows.

Category Name	Quantity	Amount
Beverages	925	\$27,761.57
Condiments	378	\$10,773.27
Confections	880	\$22,877.18
Dairy Products	581	\$13,685.32
Grains/Cereals	189	\$3,325.40
Meat/Poultry	92	\$4,083.66
Produce	351	\$13,031.20
Seafood	669	\$9,316.54
<b>Total</b>	<b>4,065</b>	<b>\$104,854.14</b>

## 7.3 Sorting, Grouping and Totaling

### 7.3.1 Sorting data

Sorting means placing data in some kind of order to help you find and evaluate it. For example, you may want to have a customer list sorted alphabetically by name or by country.

To sort your data, you may use SQL. Use the **ORDER BY** clause to have your results displayed in a sorted order.

```
SELECT EmployeeID  
,LastName  
,FirstName  
,HireDate  
FROM Employees  
ORDER BY HireDate; /* ascending sort */
```

In the example above, results will come back in ascending order by hire date. To explicitly specify ascending or descending order, add ASC or DESC, to the end of your ORDER BY clause. The following is an example of a descending

order sort.

```
ORDER BY HireDate DESC; /* descending sort */
```

### **7.3.2 Totaling**

You can sum the values, count all the values or only those values that are distinct from one another, and determine the maximum, minimum, average.

To add totals, there are two ways.

1. You can add total using (Formula) fields of Microsoft Word, such as SUM(above). For more detail information, refer to “Using Formula Fields” in this document.

2. You can use the aggregate functions in SQL statement, such as COUNT, SUM, AVG, MAX, MIN.

- (1) In the fixed table report, you can add a total directly using a separate SQL.

- (2) In the variable table report, you must add the total first using a Fixed Table report function before you use the Variable Table report function. Because the cell address of the total field will change after you use Variable Table report function.

### **7.3.3 Grouping data and Subreports**

Grouped data is data that is sorted and broken up into meaningful groups. In a customer list, for example, a group might consist of all those customers living in the same Region.

To group data in a report, you should use GROUP TABLE REPORT function or FORM REPORT function. For more detail information, refer to “GroupTableReport Method”, “FormReport Method”, “Group Table Report” and “Form Report” in this document.

Using the feature of grouping data, you can make subreports within a report.

A subreport would typically be used to perform one-to-many lookups such as

Customer / Order / OrderDetails.

To make sub reports within the main report,

1. Write a JOIN SQL statement to access data from two or more tables. For example, you can join Customers, Orders and OrderDetails tables.
2. Use GROUP TABLE REPORT function or FORM REPORT function.

For more detail information, refer to the samples invoice.wrf, product\_catalog.wrf and sales\_detail.wrf within WDRReportCom.

### **7.3.4 Subtotaling**

A subtotal is a summary that totals or sums numeric values in a group. You can sum the values in each group, count all the values in each group, and determine the maximum, minimum, average in each group. For example, determine the total sales per sales representative in a sales report.

To add subtotals, you can use aggregate functions in SQL statement.

1. Use aggregate function and GROUP BY clause, get summary data for each group, and insert results into a temporary table.
2. If you have the different kinds of summaries, repeat the step 1, and insert results into another temporary table.
3. Use group table report function or form report function, and join the detail data and the summary data using JOIN. The summary fields must be included in the group list.
4. Except for sub-totals, you can add total too using aggregate function in SQL statement. You must add total first using a Fixed Table report function before you use the Variable Table report function. Because the cell address of the total field will change after you use Variable Table report function.

For more detail information, please refer to the samples invoice.wrf and sales\_detail.wrf within WDRReportCom.

## **7.4 Pictures**

### **7.4.1 Inserting pictures into a report template**

To make eye-catching reports, you can add pictures to your reports. You can directly insert pictures into the report template in Microsoft Word. For example, you want to display a logo in your report. You can insert the logo graphics file into the report template. For more information about adding pictures to tables, refer to *Microsoft Word Help*.

### **7.4.2 Inserting pictures into a report**

Except for inserting the static pictures during report design, you may insert pictures during report building process. WDRReportCom can put the graphics files into the report, and support all graphics file format that Microsoft Word support.

To insert pictures into a report using WDRReportCom, you should do as follows:

1. Store the paths and names of the graphics files in the database

You store the paths and names of the picture files in database, do not store the pictures. The file path can be a relative path, an absolute path or a URL. For example, you store "images\emp1.jpg" in Photo field.

2. Specify the inserted way, text wrapping style and size in the report template

To specify the inserted way, text wrapping style and size, you should write a formatting expression in the report template file. For a table report, you write a formatting expression in the cell. For a form report, write a formatting expression in the field switch "\#". WDRReportCom will get the formatting expression, and insert a picture into the report according to the instruction in the format expression.

3. Write the report function in a WRF file, and identify the image fields using the IMAGE argument. For example,

```
@F1=Report(table=1 ... image=photo)
```

4. Use WDRReportCom to generate report with pictures

WDRReportCom will submit the SQL statement and get the data from database, read the graphics files according to the paths and names, and insert them into the report. If the path and file name of the picture is "", WDRReportCom will return "". WDRReportCom will return "#Error" if it does not find the file of the picture.

For more detail information about pictures, refer to the samples employee\_profile.wrf, product\_catalog.wrf within WDRReportCom.

## Chapter 8 Hints and Tips

You can run WordReport.exe in command line. The format is:

```
wordreport <wrf file name> [-d] [-u1 user1] [-p1 pwd1] ... [pa1 pa2 ...]
```

For example:

```
wordreport c:\wordreport\monthlysales.wrf 199605
```

WordReport.exe can be scheduled with Windows Scheduled Tasks or other tools. The process of generating reports can be fully automated, periodically or on events.

WDRReportCom comes with a sample database, VB sample programs, VBA sample programs and sample reports. You can use them when learning the program. To use the samples, you must add a data source named "Report Sample" to specify the sample database.

To make a report template, you can use some sample data. It is very useful especially for formatting. After you have made the report template, you delete the sample data.

For a table report, you can format the value from data sources with a format expression. You should write a format expression into a data cell in the report template file first. WDRReportCom will get the text of the cell as a format expression before it puts a value into a cell, and output the value using the format expression.

You can define the different formats and colors for positive values, negative values and zeros.

For a form report, you can format the value from data sources with the switch of a merge field or quote field like “\#” or “\@”.

In the sample file “monthly\_sales.doc”, there is a macro called “CellRef”. The macro can tell you the table number and the cell reference.

An irregular table does not have the same number of cells for each row or column. It does make it harder to process the document. In an irregular table, you have some difficulty to reference a cell, and an error may occur when you try to work with some rows or columns.

You can set the width of cells in the same column to be different, and keep the table have the same number of cells for each row. Select two or more cells in one row, merge them into one cell, and split the cell into two or more cells. You can drag the boundary and change the cell width.

To create a chart in the report template file, you can use some sample data. Using sample data, you can set the various chart options. After you have made the report template, you delete the sample data.

For MSGraph chart, on the datasheet, the leftmost column and the top row, which are commonly used for legend text or axis labels, are referred to as column 0 (zero) and row 0 (zero).

By default, Microsoft Word 2007 uses Microsoft Excel to create charts, but doesn't expose the chart as a normal Excel object. WDRReportCom can not access the charts. You must insert an Excel chart object that WDRReportCom

can access.

You can use (formula) fields to perform calculations in a report template file.

WDRReportCom is a converter too. Besides Microsoft Word document, you can generate a report in other file format such as HTML, XML, RTF, text, Works.

You also can convert data from database to other file format.

You can protect the generated report so that it can not be modified.

You can edit a WRF file (.wrf) with a text editor such as Notepad.

In a WRF file, for the report template file, report file and log file, it is possible to give a relative path. If it is a relative path, the base path is the path of the WRF file.

In a WRF file, you can use parameters in the SQL statements. To use parameters, you must define them first.

In a WRF file, you can use parameters in the paths and names of the report file, template file and log file. To use parameters, you must define them first.

You should be careful to define a unique name for each parameter, because WDRReportCom will replace all strings that are the same as the names of the parameters. It is a good choice a name begins with the "\$" character such as "\$ReportDate".

If you get some errors when you run WordReport.exe, you can check the

default log file "WordReport.log" under the WDRReportCom program directory. If you do not define the log file in the WRF file, or can not create the log file defined, you can find log information in the WordReport.log.

In the [SQL] section in the WRF file, you can use comments. A comment is the "/" characters, followed by any sequence of characters (including new lines), followed by the "\*" characters. You cannot nest comments.

To add totals or subtotals, you can use the aggregate functions in SQL statement.

To group data in a report, you should use GroupTableReport method, FormReport method or Group Table Report function, Form Report function.

In Group Table Report function or Form Report function in the WRF file, the order of groups should be in accordance with the order of ORDER BY clause in the SQL statement.

If you add a bookmark in a table, you can reference the table using the bookmark.

WDRReportCom supports nested tables. You can reference a nested table by a table index like 2-1-2 or a bookmark.

In a HTML file, you can add a bookmark using the <a> tag. For example, The following example defines a bookmark named "salesreport".

```
<a name=salesreport>Sales Report</a>
```

You can create reports with pictures using WDRReportCom. You should store the path and name of the graphics file in the database, identify the image fields in the report function, and specify the inserted way, text wrapping style and size in the report template file.

If you insert pictures into the cells using table report function, you can adjust the position of the pictures by changing the cell margins.

To convert from pixels to points, it is depend on the screen resolution (DPI). If you have a 96 dpi screen (Windows PC), 4 pixels are equal to 3 points.

SetDocVariable method and DocVariable function support headers and footers. You can use it to put data into headers or footers.

It is very useful to create a temporary table. You can prepare data using INSERT/UPDATE/DELETE/INSERT SELECT, and then make a report using REPORT function.

You can write a program to make a WRF file using C, perl or DOS shell, and then run WordReport.exe to generate report. The two steps can be written into a batch file.

It will take more time using COPYRANGE argument than using RANGE argument. Fixed table report is slower than non-group variable table report with RANGE argument, and faster than non-group variable table report with COPYRANGE argument. In general, group variable table report is slower than non-group variable table report. But if the ranges of all groups and details are same and you do not use COPYRANGE argument, it is faster.

VarTableReport, GroupTableReport, FormReport methods will use clipboard.  
You can not copy and paste during report generating.

WDRReportGen supports Microsoft Word 2007. You can use docx file as report  
file and template file. Please copy “wconv2007.cfg” to “wconv.cfg”.

## Chapter 9 Format Expressions in Data Cells

For a cell in which data are got from data source, you can set the format using a format expression. WDRReportCom gets the text from the cell, and outputs the result using it as the format expression. In fact, WDRReportCom calls the format function in Visual Basic. For more information about format, refer to Format Function in *Visual Basic for Applications Reference*.

### A.1 Formats for Numeric Values

A user-defined format expression for numbers can have from one to four sections separated by semicolons. If the format argument contains one of the named numeric formats, only one section is allowed.

If you use	The result is
One section only	The format expression applies to all values.
Two sections	The first section applies to positive values and zeros, the second to negative values.
Three sections	The first section applies to positive values, the second to negative values, and the third to zeros.
Four sections	The first section applies to positive values, the second to negative values, the third to zeros, and the fourth to Null values.

The following example has two sections: the first defines the format for positive values and zeros; the second section defines the format for negative values.

```
"$#,##0;($#,##0)"
```

If you include semicolons with nothing between them, the missing section is printed using the format of the positive value. For example, the following format displays positive and negative values using the format in the first

section and displays "Zero" if the value is zero.

"\$#,##0;;\Z\e\r\o"

The following table identifies characters you can use to create user-defined number formats:

Character	Description
None	Display the number with no formatting.
(0)	Digit placeholder. Display a digit or a zero. If the expression has a digit in the position where the 0 appears in the format string, display it; otherwise, display a zero in that position. If the number has fewer digits than there are zeros (on either side of the decimal) in the format expression, display leading or trailing zeros. If the number has more digits to the right of the decimal separator than there are zeros to the right of the decimal separator in the format expression, round the number to as many decimal places as there are zeros. If the number has more digits to the left of the decimal separator than there are zeros to the left of the decimal separator in the format expression, display the extra digits without modification.
(#)	Digit placeholder. Display a digit or nothing. If the expression has a digit in the position where the # appears in the format string, display it; otherwise, display nothing in that position. This symbol works like the 0 digit placeholder, except that leading and trailing zeros aren't displayed if the number has the same or fewer digits than there are # characters on either side of the decimal separator in the format expression.
(.)	Decimal placeholder. In some locales, a comma is used as the decimal separator. The decimal placeholder determines how many digits are displayed to the left and right of the decimal separator. If the format expression contains only number signs to the left of this symbol, numbers smaller than

	<p>1 begin with a decimal separator. To display a leading zero displayed with fractional numbers, use 0 as the first digit placeholder to the left of the decimal separator. The actual character used as a decimal placeholder in the formatted output depends on the Number Format recognized by your system.</p>
(%)	<p>Percentage placeholder. The expression is multiplied by 100. The percent character (%) is inserted in the position where it appears in the format string.</p>
(,)	<p>Thousand separator. In some locales, a period is used as a thousand separator. The thousand separator separates thousands from hundreds within a number that has four or more places to the left of the decimal separator. Standard use of the thousand separator is specified if the format contains a thousand separator surrounded by digit placeholders (0 or #). Two adjacent thousand separators or a thousand separator immediately to the left of the decimal separator (whether or not a decimal is specified) means "scale the number by dividing it by 1000, rounding as needed." For example, you can use the format string "##0,," to represent 100 million as 100. Numbers smaller than 1 million are displayed as 0. Two adjacent thousand separators in any position other than immediately to the left of the decimal separator are treated simply as specifying the use of a thousand separator. The actual character used as the thousand separator in the formatted output depends on the Number Format recognized by your system.</p>
(:)	<p>Time separator. In some locales, other characters may be used to represent the time separator. The time separator separates hours, minutes, and seconds when time values are formatted. The actual character used as the time separator in</p>

	formatted output is determined by your system settings.
(/)	Date separator. In some locales, other characters may be used to represent the date separator. The date separator separates the day, month, and year when date values are formatted. The actual character used as the date separator in formatted output is determined by your system settings.
(E- E+ e- e+)	Scientific format. If the format expression contains at least one digit placeholder (0 or #) to the right of E-, E+, e-, or e+, the number is displayed in scientific format and E or e is inserted between the number and its exponent. The number of digit placeholders to the right determines the number of digits in the exponent. Use E- or e- to place a minus sign next to negative exponents. Use E+ or e+ to place a minus sign next to negative exponents and a plus sign next to positive exponents.
- + \$ ( )	Display a literal character. To display a character other than one of those listed, precede it with a backslash (\) or enclose it in double quotation marks (" ").
(\)	Display the next character in the format string. To display a character that has special meaning as a literal character, precede it with a backslash (\). The backslash itself isn't displayed. Using a backslash is the same as enclosing the next character in double quotation marks. To display a backslash, use two backslashes (\\).  Examples of characters that can't be displayed as literal characters are the date-formatting and time-formatting characters (a, c, d, h, m, n, p, q, s, t, w, y, / and :), the numeric-formatting characters (#, 0, %, E, e, comma, and period), and the string-formatting characters (@, &, <, >, and !).
("ABC")	Display the string inside the double quotation marks (" "). To

	include a string in format from within code, you must use Chr(34) to enclose the text (34 is the character code for a quotation mark (")).
--	--

## A.2 Formats for String Values

A format expression for strings can have one section or two sections separated by a semicolon (;).

If you use	The result is
One section only	The format applies to all string data.
Two sections	The first section applies to string data, the second to Null values and zero-length strings ("").

You can use any of the following characters to create a format expression for strings:

Character	Description
@	Character placeholder. Display a character or a space. If the string has a character in the position where the at symbol (@) appears in the format string, display it; otherwise, display a space in that position. Placeholders are filled from right to left unless there is an exclamation point character (!) in the format string.
&	Character placeholder. Display a character or nothing. If the string has a character in the position where the ampersand (&) appears, display it; otherwise, display nothing. Placeholders are filled from right to left unless there is an exclamation point character (!) in the format string.
<	Force lowercase. Display all characters in lowercase format.
>	Force uppercase. Display all characters in uppercase format.
!	Force left to right fill of placeholders. The default is to fill placeholders from right to left.

## A.3 Formats for Date/Time Values

The following table identifies characters you can use to create user-defined date/time formats:

Character	Description
(:)	Time separator. In some locales, other characters may be used to represent the time separator. The time separator separates hours, minutes, and seconds when time values are formatted. The actual character used as the time separator in formatted output is determined by your system settings.
(/)	Date separator. In some locales, other characters may be used to represent the date separator. The date separator separates the day, month, and year when date values are formatted. The actual character used as the date separator in formatted output is determined by your system settings.
c	Display the date as ddddd and display the time as tttt, in that order. Display only date information if there is no fractional part to the date serial number; display only time information if there is no integer portion.
d	Display the day as a number without a leading zero (1 – 31).
dd	Display the day as a number with a leading zero (01 – 31).
ddd	Display the day as an abbreviation (Sun – Sat).
dddd	Display the day as a full name (Sunday – Saturday).
ddddd	Display the date as a complete date (including day, month, and year), formatted according to your system's short date format setting. The default short date format is m/d/yy.
dddddd	Display a date serial number as a complete date (including day, month, and year) formatted according to the long date setting recognized by your system. The default long date format is mmmm dd, yyyy.
aaaa	The same as dddd, only it's the localized version of the string.

w	Display the day of the week as a number (1 for Sunday through 7 for Saturday).
ww	Display the week of the year as a number (1 – 54).
m	Display the month as a number without a leading zero (1 – 12). If m immediately follows h or hh, the minute rather than the month is displayed.
mm	Display the month as a number with a leading zero (01 – 12). If m immediately follows h or hh, the minute rather than the month is displayed.
mmm	Display the month as an abbreviation (Jan – Dec).
mmmm	Display the month as a full month name (January – December).
oooo	The same as mmmm, only it's the localized version of the string.
q	Display the quarter of the year as a number (1 – 4).
y	Display the day of the year as a number (1 – 366).
yy	Display the year as a 2-digit number (00 – 99).
yyyy	Display the year as a 4-digit number (100 – 9999).
h	Display the hour as a number without leading zeros (0 – 23).
Hh	Display the hour as a number with leading zeros (00 – 23).
N	Display the minute as a number without leading zeros (0 – 59).
Nn	Display the minute as a number with leading zeros (00 – 59).
S	Display the second as a number without leading zeros (0 – 59).
Ss	Display the second as a number with leading zeros (00 – 59).
t t t t t	Display a time as a complete time (including hour, minute, and second), formatted using the time separator defined by the time format recognized by your system. A leading zero is displayed if the leading zero option is selected and the time is before 10:00 A.M. or P.M. The default time format is h:mm:ss.
AM/PM	Use the 12-hour clock and display an uppercase AM with any hour before noon; display an uppercase PM with any hour between noon and 11:59 P.M.
am/pm	Use the 12-hour clock and display a lowercase AM with any

	hour before noon; display a lowercase PM with any hour between noon and 11:59 P.M.
A/P	Use the 12-hour clock and display an uppercase A with any hour before noon; display an uppercase P with any hour between noon and 11:59 P.M.
a/p	Use the 12-hour clock and display a lowercase A with any hour before noon; display a lowercase P with any hour between noon and 11:59 P.M.
AMPM	Use the 12-hour clock and display the AM string literal as defined by your system with any hour before noon; display the PM string literal as defined by your system with any hour between noon and 11:59 P.M. AMPM can be either uppercase or lowercase, but the case of the string displayed matches the string as defined by your system settings. The default format is AM/PM.

# Chapter 10 License and Support

## 10.1 License

### Your Agreement to This License

You should carefully read the following terms and conditions before using, installing, copying, or distributing this software. Your use, installation, copying, or distribution of WDRReportCom indicates your acceptance of this agreement ("License").

### **NO WARRANTY**

WDRReportCom IS DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.

### **Evaluation License**

WDRReportCom is not free software. You may use this software for evaluation purposes without charge for a period of 30 days. If you use this software after the 30 day evaluation period, you must purchase it.

You may copy the evaluation version of this software and documentation as you wish, and give exact copies of the original evaluation version to anyone, and distribute the evaluation version of the software and documentation in its unmodified form via electronic means. You are specifically prohibited from charging, or requesting donations without permission from the author.

## **Developer License**

The software is licensed per developer. This means that each developer using the software needs one license. The developer may use the software on one or more computers. You may develop your application that bundles or makes use of the software directly/indirectly. You can not use the software to build competitive products of any kind, like WDRReportGen.

You may not resell, rent, lease, sub-license or distribute the software alone. The software must be distributed as a component of an application and bundled with an application or with the application's installation files. You may distribute royalty-free the run-time files of the software with your applications. You need to duly inform your customers that they are not allowed to use the software independently from your application.

## **10.2 Technical Support**

If you encounter any problems in usage of WDRReportCom, and need the technical support:

- Go to our support web site at:  
<http://www.ljzsoft.com/support.htm>
- Send email to [support@ljzsoft.com](mailto:support@ljzsoft.com)