WDReportGen

User Manual

Version 5.1

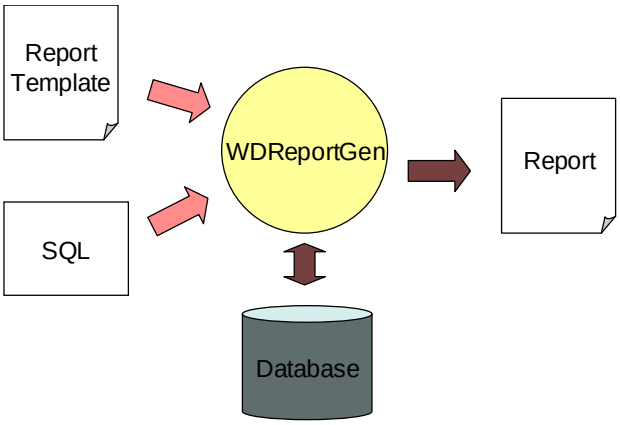
2014-4

Contents

Chapter 1 Introduction

1.1 Overview

WDReportGen is a report generator for Microsoft Word that outputs reports in Microsoft Word document format. If you know how to use Microsoft Word and write SQL statements, you can use WDReportGen to create all kinds of reports as you need.



To create a report, WDReportGen need to read a report template file and a WRF file. The report template file is a Microsoft Word document that defines the layouts and formats of a report. The WRF file contains SQL statements and some information, and tells WDReportGen how to get data from database and how to put data into a report. First WDReportGen creates a blank report using the report template file, executes SQL statements in the WRF file, and puts the data from database into the report.

1.2 Features

WDReportGen 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 document 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 WDReportGen is based on the fact that all formatting is done directly in Microsoft Word. You can take advantage of Microsoft Word including text formatting, tables, pictures and graphics, drawing, chart, page setup, headers and footers, and more.

- Manipulating Word reports with OpenXML SDK

Open XML is an open file format for the core document-oriented Office applications. It facilitates document creation and manipulation in server environments where you do not need to install the Office client applications.

- Accessing to data using SQL

WDReportGen 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.

- Creating reports without programming experience

You know how to use Microsoft Word and how to write SQL, it is enough. It does not require programming to create reports.

- Connection to databases using ADO.Net

Using ADO.NET to connect to databases, WDReportGen is independent of database systems. It can access to almost all of databases such as Oracle, DB2, Sybase, Informix, Microsoft SQL Server, Teradata, MySQL, Microsoft Access and dBase through OLE DB and ODBC.

- Supporting multi-databases in one report

WDReportGen supports multi-databases in one report. You can get data from some different databases such as Oracle, DB2 and Microsoft SQL Server, and put these data into one report.

- Generating reports with parameters

WDReportGen enables you to create reports with parameters. You may use parameters in SQL statements. You will be asked to input the values of parameters while creating reports.

- Supporting Windows mode and command line mode

WDReportGen supports command line mode. So it is possible to call WDReportGen from other program. For developers, you can integrate WDReportGen into your application.

- Creating complex reports

You can create complex reports. The complexity might come from report formatting as well as report content.

- Creating reports with charts

WDReportGen enables you to include sophisticated, colorful charts in your reports. You can use charts any time you want to improve the usefulness of a report.

- Creating reports with pictures

WDReportGen can insert pictures from the graphics files, and set the inserted way, text wrapping style and size of the pictures according to your instruction.

- Many reports in one Microsoft Word document

One Microsoft Word document may contain many reports. You can generate a book of reports in one generating process.

- Generating reports automatically

The process of report generation can be fully automated, periodically or on events. WDReportGen can be scheduled with Windows Scheduled Tasks or other tools.

- One time configuration

With on time configuration, you can repeatedly generate reports especially periodic reports such as daily, weekly, monthly and annual reports.

- Flexible deployment

WDReportGen can be run on your desktop or server.

Chapter 2 Installation and Startup

2.1 Software Requirements

Supported Operating System:

- Microsoft Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 or later.

This software requires the following:

- Microsoft .NET Framework 4.
- Open XML SDK 2.5 for Microsoft Office.

2.2 Installing WDReportGen

Run the installation program, and follow the instructions to complete WDReportGen installation.

If you don't have Microsoft .NET Framework 4 installed, please install it first.

If you don't have Open XML SDK 2.5 installed, please install it first.

2.3 Uninstalling WDReportGen

1. Quit WDReportGen.

2. Double-click the **Add/Remove Programs** icon in the Windows Control Panel.

3. Click **WDReportGen** in the **Currently installed programs** box, and then click the **Change/Remove** button.

4. Follow the instructions on the screen to complete uninstalling the program.

2.4 Command Line

WDReportGen can be run in Windows mode or command line mode. The

Syntax of command is:

wordreport <wrf file name> [-C] [-D] [-I interval] [-U1 user1] [-P1 pwd1] … [-U10 user10] [-P10 pwd10] [pa1 pa2 … pa10]

wrf file name	Specifying a WRF (.wrf) file that tells WDReportGen how to get data from data sources and how to put data into a report.
-C	Run WDReportGen in command line mode.
-D	Display the generated report with Microsoft Word.
-I interval	Log the processing records message. If interval is greater than 1, it is the interval of records. If interval is less than 1, it is the percent of interval.
-U1 user1 … -U10 user10	Specify the user names. user1 is the user name of the first data source. user2 is the user name of the second data source……
-P1 pwd1 ... -P10 pwd10	Specify the passwords. pwd1 is the password of the first data source. pwd2 is the password of the second data source……
pa1 … pa10	The values of the parameters defined in the WRF file. You can use parameters in SQL statements. WDReportGen will replace the names of parameters in a SQL statement with the actual values before it executes a 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 sales date, and the second is the category of the products. You can run WDReportGen in command line mode as follows:

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

Chapter 3 Quick Start

3.1 Learning how to use WDReportGen

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

- You can study the sample reports and sample database included with WDReportGen.
- You can use the detailed descriptions and instructions in the “My First Report”.

3.2 Sample Database

WDReportGen 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 WDReportGen, 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. Double-click **Administrative Tools**, and then double-click **Data Sources (ODBC)**. The **ODBC Data Source Administrator** dialog box appears.
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 Steps of Reporting

To create a report with WDReportGen, you should do as follows:

1. Prepare works

Before you create a report, you should determine the layout of report, and know where and how to get the data.

You must know how to access to the databases you are reporting from. So you need the data source name, user name and password. If you don't have data sources added, please add data sources first. Run ODBC Administrator, you can add a new data source. For detailed information about configuring ODBC, refer to *ODBC Administrator Help*.

2. Make a report template file

Create a report template file using Microsoft Word. For detailed information about report template, refer to "Report Templates" in this document.

3. Create a WRF file

Create a WRF file with a .wrf extension using WDReportGen. There are two steps to create a WRF file.

(1) Configure the report

Define the names of data sources, the name of report template file, the name of report file and the name of log file. If you want to use parameters in SQL statements, define these parameters.

(2) Write functions

Write functions and SQL statements that specify how to get data from data sources and how to put data into the report.

For detailed information, refer to “Reporting with WDReportGen” in this document.

Define the names of data sources, the name of report template file, the name of report file and the name of log file. If you want to use parameters in SQL statements, define these parameters.

(2) Write functions

Write the functions and SQL statements that specify how to get data from data sources and how to put data into the report.

4. Run the WRF file

Run the WRF file to generate a report in Microsoft Word document. For detailed information about running report, refer to “Running a WRF File” in this document.

3.4 My First Report

The following tutorial has been designed to guide you to create your first report. In this tutorial, you will get an introduction to the program as you create a Customer List report. The Customer List is one of the most basic business reports and typically has information such as Customer Name, City, Country, and Contact Name.

3.4.1 Creating a report template

1. Run Microsoft Word, a new document will open.

2. On the **Table** menu, point to **Insert**, and then click **Table**. Under **Table size**, select the number of columns and rows. Press **OK** button.

3. Click the cell A1, type “Customer Name”. In the same way, you input “City”, “Country” and “Contact Name” into the cells B1, C1 and D1.

4. Format the text of A1, B1, C1 and D1 as you like, including font, font size, font colour, bold, background, alignment and border.

5. You can change the width of these columns. The report template you have made is as follows:

Customer Name	City	Country	Contact Name

6. Click **Save** on the **File** menu, chose a directory such as “C:\Report”, type custlist.docx in the **File name** box and press **Save** button.

7. Click **Close** on the **File** menu.

3.4.2 Creating a WRF file

1. Run WDReportGen.

2. Click **New** on the **File** menu.

3. Click **Save** on the **File** menu, chose the directory to which you have saved the report template, type custlist.wrf in the **File name** box and press **Save** button.

3.4.3 Configuring the report

1. On the **Report** menu, click **Configuration**. The **Configuration** dialog box appears.

2. Click the **File** tab.

In the **Template File** box, type custlist.docx; In the **Report File** box, type Report\custlist.docx; In the **Log File** box, type Log\custlist.log.

3. Click the **Data Source** tab.

Press **New** button, the **New Data Source** dialog box appears. In the **Name** box, type Report Sample, press **OK** button.

4. On the **Configuration** dialog box, press **OK** button.

3.4.4 Inputting a function

In the editor windows, input a function as follows:

```
@F1=Report(table=1 cell=A2)
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
;
```

You can test the SQL statement in a query tool such as Microsoft Access or Microsoft Query.

3.4.5 Understanding the function

Before going any further, let us understand this function.

1. The **Report** function will execute the SQL statement, get data from data source, and put data into the report.
2. The **table** argument identifies a table, and the value 1 is the index number of the table. So it is the first table.
3. The **cell** argument specifies the cells that the first record will be filled into. The value is A2. So WDReportGen will fetch the first record, put the value of CompanyName field into A2, the value of CityName field into B2, the value of CountryName field into C2, and the value of ContactName field into D2. And then it will fetch the next record, put them into A3,B3,C3 and D3……

3.4.6 Running a WRF file

1. On the **Report** menu, click **Run**, the **Run Report** dialog box appears.

2. Press **Start** button to run the WRF file.

3. WDReportGen will generate a report.

4. After the status is **Done**, click **Close** button.

3.4.7 Opening a report

1. On the **File** menu, click **Open Report File** to open the report you have generated.

You can view and check the report.

2. On the **File** menu, click **Open Log File** to open the log file that recorded the log information in the report generating. You can check the log.

3. Close the report file and the log file.

3.4.8 Modifying the report template

1. On the **File** menu, click **Open Template File** to open the report template.

2. Change the width of columns. It is very useful to copy some sample data from the report file into the report template for formatting.

3. Insert text before the table, and type Customer List as the report title. To insert text before a table, click in the upper-left cell in the first row of the table, place the insertion point before the text, and then press **ENTER**.

4. Select the second row, and insert a row to the table.

5. Add a border to the table. Select the table, click **Borders and Shading** on the **Format** menu, and then click the **Borders** tab. Select the options you want, and press **OK** button. The external border can be different from the internal border. The report template you have made is as follows:

Customer List

Customer Name	City	Country	Contact Name

6. Select the first row of the table, and click **Heading Rows Repeat** on the **Table** menu.

7. Save and close the template file.

3.4.9 Modifying the function

In the editor windows, modify the function as follows:

```
@F1=Report(table=1 cell=A2 rangecount =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
;
```

The **rangecount** argument specifies the number of ranges in the report template. You have defined two blank ranges in the report template. One row is one range. If you hope that the format of the last row/column border is different from the others, you can define two ranges in the report template.

3.4.10 Generating the report again

1. Save the WRF file.

2. Run the WRF file to generate the report.

3. Open the report, view and check the report.

The report should now look similar to the following:

Customer List

Customer Name	City	Country	Contact Name
Alfreds Futterkiste	Berlin	Germany	Maria Anders
Ana Trujillo Emparedados y helados	México D.F.	Mexico	Ana Trujillo
Antonio Moreno Taquería	México D.F.	Mexico	Antonio Moreno
Around the Horn	London	UK	Thomas Hardy
Berglunds snabbköp	Luleå	Sweden	Christina Berglund
Blauer See Delikatessen	Mannheim	Germany	Hanna Moos
Blondel père et fils	Strasbourg	France	Frédérique Citeaux

Now you have created a report.

3.5 Samples

After WDReportGen is installed, some sample reports are installed too. Use these reports to learn WDReportGen. The sample reports can be changed to adapt to your own needs.

The sample reports include a sample database, some report template files (.docx) and WRF files (.wrf). They are located in the Application Data\LJZsoft under All Users or your profile folder.

Directory	Description
{commonappdata}\LJZsoft\Common\SampleDatabase	Contains the sample database “Sample.mdb”.
{commonappdata}\LJZsoft\WDReportGen\Samples	Contains the report template files (.docx) and the WRF files (.wrf).
{commonappdata}\LJZsoft\WDReportGen\Samples\Report	Contains the report files (.docx) generated by WDReportGen.
{commonappdata}\LJZsoft\WDReportGen\Samples\Log	Contains the log files created by WDReportGen during generating report files.

{commonappdata} is the path to the Application Data folder under All Users.

The Application Data folder is usually at:

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

Windows Vista or later: C:\ProgramData\

Chapter 4 Report Templates

4.1 About Reports

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

4.2 About Report Templates

To make a report using WDReportGen, you should create a report template first. The report template is a Microsoft Word document that defines the layouts, formats and styles of 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 tables you will fill data into.

WDReportGen will generate the report based on the report template file. All static contents, 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 .docx. 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 Ranges

A range represents a contiguous area in a document. Each range is defined by a starting and ending character position. In a table, a range represents a cell, a row, a column, or a selection of cells containing one or more contiguous blocks of cells.

4.3.5 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.

4.3.6 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.

4.3.7 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.

4.3.8 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.

4.3.9 Charts

Charts are used to display series of numeric data in a graphical format to make

it easier to understand large quantities of data and the relationship between different series of data.

A chart has many elements: chart area, plot area, data points, data series, axis, legend, title, and data label. Some of these elements are displayed by default, others can be added as needed. You can change the display of the chart elements by moving them to other locations in the chart, resizing them, or by changing the format. You can also remove chart elements that you do not want to display.

4.3.10 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.

4.3.11 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.

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

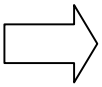
Variable table report: The number of rows or columns in the table is unfixed, and it is variable as the number of result records. When WDReportGen executes a SQL statement, it repeats the table rows or columns for each record or group, and then puts data into cells of the table.

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

	A	B
1		
2		
3		

The fixed table defined in the report template file



	A	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 do not know how many records will return from database. You need to create a table in the report template file, and make one or two blank ranges. WDReportGen will add ranges according to the number of records returned from data source.

Date	Item Id	Sales

Date	Item Id	Sales
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

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 define two blank ranges.

The ranges can be nested. The inside range is used for the detail data, and the external range is used for the group. WDReportGen 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. WDReportGen 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, WDReportGen calls the format function in Visual Basic. The text got from a cell is used as the format expression in format function. 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.

4.4.7 Referencing Tables

You can reference a table by an index number or a bookmark. The index number represents the position of 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 WDReportGen supports is 3.

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 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**.

4.4.8 Formatting Cells for Pictures

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

You should store the path and name of graphics files in the database, and identify the image fields in the report function. WDReportGen will read the graphics files, and insert them into the cells in the report file. You can adjust the position of 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. WDReportGen will get the text of 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
BEHIND	Behind 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 picture is set to 100 points. “H50” means that the height of picture is set to 50 points. The default means the original size.

If you just specify the width or height of picture, not both, WDReportGen will retain the original proportions of picture when WDReportGen resize it.

Example

inline w120 h90

Remarks

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

4.5 Form Reports

4.5.1 About Form Reports

Beside table reports, WDReportGen 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:

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 WDReportGen 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 WDReportGen 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»

The ranges can be nested. The inside range is used for the detail data, and the external range is used for the group. For example, you define a bookmark Category for the product category, and a bookmark Product for the product.

[«CategoryName»

«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. WDReportGen supports many popular graphics file formats: bitmap, JPG, GIF, PNG, TIFF and so on. For the graphics file formats WDReportGen supports, refer to *Microsoft Word Help*.

You should store the path and name of graphics files in the database, and

identify the image fields in the report function. WDReportGen 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. WDReportGen 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
BEHIND	Behind text-wrapping style

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

Example

\# “square w84”

Remarks

On the supposition that the original picture is size 144 x 168 points. WDReportGen will insert a floating picture, apply the square text-wrapping

style, set the height of picture to 72 points, and the width to 84 points.

4.6 Charts

4.6.1 About Charts

Charts are visually appealing and make it easy for users to see comparisons, patterns, and trends in data. Chart tools are fully integrated with Microsoft Office 2007. For Microsoft Word 2007 or later, the default chart software is Microsoft Excel.

Charts that you create will be embedded in Microsoft Word, and the chart data is stored in an Excel worksheet that is incorporated in the Word file.

4.6.2 Creating a Blank Chart using Microsoft Excel

To create an Excel chart in the report using WDReportGen, you need to add an Excel 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.

To add an Excel chart in the template file in Microsoft Word 2007 or later:

1. Open the report template file using Microsoft Word.
2. Place the insertion point where you want to create the chart. On the **Insert** tab, in the **Illustrations** group, click **Chart**.
3. In the **Insert Chart** dialog box, click a chart, and then click **OK** button.

Microsoft Excel will open and display sample data on a worksheet. 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, you should define the blank ranges according to number of records.

- If the report type is var, you should define one or two blank ranges.

5. After you have finished the modification, delete data from the worksheet of chart. You should keep a blank chart in the report template file. WDReportGen will put data into the worksheet of chart.

6. When you've finished, click the **Microsoft Office Button** in Excel, and then click **Close**.

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

4.6.3 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 Reporting with WDReportGen

5.1 Creating and Opening WRF Files

5.1.1 About WRF files

To generate a report with WDReportGen, you must create a WRF file with a .wrf extension. The WRF file contains information such as the name of report template file, the name of report file, log file name, data sources, parameters and functions. The WRF file tells WDReportGen how to get data from data sources and how to put data into a report.

5.1.2 Create a new WRF file

On the **File** menu, click **New**.

5.1.3 Open a WRF file

1. On the **File** menu, click **Open**.
2. In the **Look in** list, click the drive, folder, or Internet location that contains the file you want to open.
3. In the folder list, locate and open the folder that contains the file.
4. Click the file, and then press **Open** button.

5.1.4 Save a WRF file

On the **File** menu, click **Save**. If you're saving the file for the first time, you'll be asked to give it a name.

If you want save a file to another name, do as follows:

1. On the **File** menu, click **Save As**.
2. In the **File name** box, enter a new name for the file.

3. Press **Save** button.

5.2 Configuring Files

5.2.1 About files

You should specify the report template file, report file, report file type and log file. The report template file defines the layouts, formats and styles of report. The report file is the report you want to generate. The type of report file can be different from the template file. The log file records the log information in the report generating.

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 WRF file. In the paths and names of report file, template file and log file, you can use parameters. For detailed information about parameters, refer to “Configuring Parameters” in this document.

5.2.2 Configuring file information

1. On the **Report** menu, click **Configuration**. The **Configuration** dialog box appears.
2. Click the **File** tab.
3. Input the path and name of template file, report file and log file into their text box. The template file and report file have a .docx extension.
4. WDReportGen 5 can not convert a file to other file format. You cannot change the **File Type** box.
5. If you want to protect the report, select the **Protect Report** check box, and input a password in the **Password** box. If the check box is selected, the Word report generated is protected, and cannot be modified without the password.
6. Press **OK** button to confirm the changes, press **Cancel** button to discard the changes.

5.2.3 Converting files

WDReportGen 5 can not convert a file to other file format.

5.3 Configuring Data Sources

5.3.1 About data sources

A data source identifies a database you want to access. WDReportGen can access to almost all of the databases such as Oracle, DB2, Sybase, Informix, Microsoft SQL Server, Teradata, MySQL, Microsoft Access and dBase through OLE DB and ODBC. It supports more than one data sources in one report. You can get data from the different databases such as Oracle, DB2 and Microsoft SQL Server, and put them into one report.

You can define a connection to a data source using an ODBC data source name or a connection string. If you use an ODBC data source name to make a connection, you should specify a user name and a password. If you use a connection string to make a connection, you also should specify a data source name that you can reference in functions.

5.3.2 Adding, modifying and deleting a data source

1. On the **Report** menu, click **Configuration**. The **Configuration** dialog box appears.
2. Click the **Data Source** tab.
3. If you want to add a data source, press **New** button, the **New Data Source** dialog box appears.

- To define a connection using an ODBC data source name, click **Using ODBC data source name** option, input data source name, user name and password, press **OK** button.

- To define a connection using a connection string, click **Using connection string** option, input data source name and connection string, select a data provider, press **OK** button.

4. If you want to modify a data source, click the data source name in the **Data Source** list box, and press **Edit** button, the **Edit Data Source** dialog box appears.

- To define a connection using an ODBC data source name, click **Using ODBC data source name** option, change data source name, user name and password, press **OK** button.
- To define a connection using a connection string, click **Using connection string** option, change data source name, connection string, and data provider, press **OK** button.

5. If you want to delete a data source, click the data source name in the **Data Source** list box, and press **Delete** button, the confirmation dialog box appears. Press **Yes** button to delete the data source.

6. You can test a data source. Click the data source name in the **Data Source** list box, and Press **Test** button to display the information of connection to the data source.

7. Select or clear the **Encrypt Password** check box. If the check box is selected, passwords will be saved in an encrypted format. Or passwords will be saved in plain text.

8. Press **OK** button to confirm the changes, press **Cancel** button to discard the changes.

5.4 Configuring Parameters

5.4.1 About parameters

You can use parameters in SQL statements. These values need to be

provided to WDReportGen before it executes these SQL statements. To use a parameter, you must declare it first. When WDReportGen generate a report, it will prompt you to input the value of the parameter. WDReportGen will replace the parameter name in the SQL statements with the actual value before it submits the SQL statements to data sources.

A parameter has a name, a title and a default value. The name of parameter identifies the parameter. You can use the names in SQL statements. The titles will be displayed in the prompt dialog box when WDReportGen is run.

Note: WDReportGen 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 “$” character. For example, you give the name “$ReportDate” for a parameter. Parameters are case-sensitive.

5.4.2 Adding, modifying and deleting a parameter

1. On the **Report** menu, click **Configuration**. The **Configuration** dialog box appears.
2. Click the **Parameter** tab.
3. If you want to add a parameter, press **New** button, the **New Parameter** dialog box appears. Input parameter name, parameter title and default value, press **OK** button.
4. If you want to modify a parameter, click the parameter name in the **Parameter** list box, and press **Edit** button, the **Edit Parameter** dialog box appears. Change the name, title and default value of the parameter, press **OK** button.
5. If you want to delete a parameter, click the parameter name in the **Parameter** list box, and press **Delete** button, the confirmation dialog box appears. Press **Yes** button to delete the parameter.

6. Press **OK** button to confirm the changes, press **Cancel** button to discard the changes.

5.5 Inputting Functions

You should input functions in the editor window. A function includes a SQL statement and some arguments. WDReportGen executes the SQL statement, and determines whether or how to add data into the report. WDReportGen sequentially executes the functions.

Each function is begin with the “@” character. Syntax:

@functionno=functionname(arguments)

sqlstatement

The *functionno* is the label of report function.

The *functionname* represents a report function.

The *arguments* for a function define various properties for the function. For example, the “table” argument identifies a table in the Microsoft Word document. An argument takes the form *Name=“Value”*. The argument value can be delimited by single or double quotes.

The *sqlstatement* is a SQL statement.

For more detailed information about functions, see “Function Reference” in this document.

You can use comments in text. A comment is the “/*” characters, followed by any sequence of characters (including new lines), followed by the “*/” characters. You cannot nest comments.

5.6 Running WRF Files

You can run a WRF file to generate a report in Microsoft Word document

format. WDReportGen supports Windows mode and command line mode.

5.6.1 Windows mode

1. On the **Report** menu, click **Run**, the **Run Report** dialog box appears.
2. If you want to display the generated report, select the **Display Report with Microsoft Word** check box.
3. Press **Start** button to run the WRF file.
4. If parameters are defined in the WRF file, WDReportGen will pop up a prompt dialog box. Input the values of parameters, and press **OK** button.
5. While WDReportGen is being run, it will display some information such as status, SQL count, error count, function No., records count and log information.
6. You can interrupt the running. Click **End** button to interrupt it. WDReportGen will immediately save and close the report.
7. Click **Close** button after completion.
8. If you want to open the report, click **Open Report File** on the **File** menu.
9. If you want to check the log, click **Open Log File** on the **File** menu.

5.6.2 Command line mode

You can run a WRF file in command line. You have defined two parameters in the WRF file “myreport.wrf”. The first parameter is sales date “$SalesDate”, and the second is the category of the products “$Category”. You can run WDReportGen in command line mode as follows:

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

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

5.7 Sorting, Grouping and Totaling

5.7.1 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 can 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 */
```

5.7.2 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, you can use the aggregate functions in SQL statement, such as COUNT, SUM, AVG, MAX, and MIN.

(1) In the fixed table report, you can add 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 the variable table report function.

5.7.3 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 argument in the REPORT function. For more detail information, refer to “Table Report Function” and “Form Report Function” in this document.

Using the feature of grouping data, you can make sub reports within a report. A sub report 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 get data from two or more tables. For example, you can join Customers, Orders and OrderDetails tables.
2. Use GROUP argument in the REPORT function.

For more detail information, refer to the samples invoice.wrf, product_catalog.wrf and sales_detail.wrf within WDReportGen.

5.7.4 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 function 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 the variable table report function, and join the detail data and the summary data using JOIN. The summary fields must be included in the group list.

For more information, refer to the samples invoice.wrf and sales_detail.wrf within WDReportGen.

5.8 Pictures

5.8.1 Inserting pictures into a report template

To make eye-catching reports, you can add pictures to your reports. You can insert pictures into the report template directly 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 documents, refer to *Microsoft Word Help*.

5.8.2 Inserting pictures into a report

Except for inserting the static pictures during report design, you want to insert pictures during report building process. WDReportGen can insert pictures from the graphics files, and support all graphics file format that Microsoft Word support.

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

1. Store the path and name of graphics files in the database

You stored the path and file name of pictures in database, did 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. Identify the image fields in the report function

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

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

3. 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 “\#”. WDReportGen will get the formatting expression, and insert a picture into the report according to the instruction in the format expression.

4. Run WDReportGen to generate report with pictures

During report generating process, WDReportGen will read the graphics files, and insert them into the report according to your instruction. If the path and file name of picture is ““, WDReportGen will return ““. WDReportGen will return “#Error” if it does not find the file of picture.

For more detail information about pictures, refer to the samples employee_profile.wrf, product_catalog.wrf within WDReportGen.

5.9 Using Parameters

To use a parameter, you must define it first. If you have defined a parameter name, you can use it in SQL statements. When WDReportGen is run, it will replace the parameter name in the SQL statements with the actual value before it submits the SQL statements to data sources. Besides in SQL statements, you can use parameters in the paths and names of report file and log file.

In fact, WDReportGen will replace all strings that are the same as the names of

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 products. The field OrderDate is the date type, and CategoryName is the 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 report file and 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_$CustomerID.docx
LogFileName=log\report_$CustomerID.log
```

or

```
ReportFileName=report\$CustomerID\report.docx
LogFileName=log\$CustomerID\report.log
```

5.10 Programming

5.10.1 Making WRF files programmatically

Sometimes you want to make a WRF file programmatically. You can do this because the WRF file is a text file. You can write a program to make a WRF file

using C, perl or DOS shell, and then run WDReportGen 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 WDReportGen in command line mode.

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 WDReportGen 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
changewrf %1 <“template.txt” >“template.wrf”
WordReport “template.wrf” –C %1
```

Chapter 6 Function Reference

6.1 Table Report Function

The TABLE REPORT function executes a SQL statement to get data from data source, and puts data into a table in the report file.

Syntax

Report(…)

sqlstatement

Arguments

TYPE = *reporttype*

TABLE = *table*

FILLORDER = *fillorder*

CELL= *celllist*

RANGE = *range*

GROUP= *grouplist*

GROUPRANGE = *grouprange*

IMAGE = *fieldlist*

RANGECOUNT = *rangecount*

PAGEBREAK = *pagelength*

NODATA = *nodataoption*

CONNECT = *datasource*

- The **TYPE** argument specifies the report type. Possible values are fix or var. “fix” means a fixed table report, and “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 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 WDReportGen supports is 3.

- The **FILLORDER** argument specifies the order in which WDReportGen fills data. Possible values are row or col. “row” means to fill data by rows. “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. It 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 …… WDReportGen 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. WDReportGen will skip or repeat the range for each record. You can reference a range of cells like “2:4” or “B2:D5”. The range “2:4” means 3 entire rows. The range “A:B” means 2 entire columns. The default range is the area that includes all cells for the records. For the fixed table report, WDReportGen will skip the range for each record. For the variable table report, it will insert the blank rows/columns for each record. The COPYRANGE argument is similar to the Range. It will copy the original range to the range where data will be filled for each record. If there is no range or copyrange argument, the default for the fixed table report is range, and the default for the variable table report is copyrange.
- The **GROUP** argument specifies the group of 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 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 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. WDReportGen will repeat the group range for each group. The range of group should contain the range of details and the area that includes all cells for this group. You reference a group range like “2:4” or “B2:D5”. For example, there are two groups, the range of group one contains all cells for the group one and the range of group two, and the range of group two contains all cells for the group two and the range of 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 or index number of field, but not simultaneously. In data source, you stored the path and file name of 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 report template file.
- The **RANGECOUNT** argument specifies the number of blank range which you defined in the report template. It is valid when the type is “var”. Possible values are 1 or 2. One means one blank range you defined, and two means two ranges. If you hope that the format of the last row/column border can be different from the others, you can define two blank ranges. Default is 1.
- 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 WDReportGen 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 no page break.

- The **NODATA** argument specifies an option when no data are returned from data source. It is valid when the type is “var”. If the value is “delrange”, WDReportGen will delete the range when no data are returned. If the value is “deltable”, WDReportGen will delete the table 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 1, Fixed Table Report

The following function makes the Word 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
;
```

Result

The fixed table report defined in the report template:

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%

The fixed table report generated in 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%

Remarks

1. The SQL statement will get the information of top 5 employees for sales, including employee name, quantity of products, and sales amount.

2. type="fix". It is a fixed table report.

3. table = 6. WDReportGen will put data into the sixth table in the report file.

4. cell=B2. The cells corresponding to the first record are "B2,C2,D2,E2".

5. The default range is "B2:E2".

6. WDReportGen executes the SQL statement, and gets data from data source. It puts the data into the report by records.

Example 2, Variable Table Report

The following function will makes the Word 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
;
```

Result

The variable table report defined in the report template:

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

The variable table report generated in the report:

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

Remarks

1. The SQL statement will get the information of customers including company name, address, city name, country name, and postal code.

2. type="var". It is a variable table report.

3. table=1. WDReportGen will put data into the first table in the report file.

4. cell=B7,B8,B9,B10. These cells correspond to the first record.

5. copyrange=1:11. Because the default range is "B7:B9", you must specify a range explicitly. WDReportGen will copy the range for each record.

6. pagebreak = 4r. WDReportGen will add a page break per 4 records.

7. WDReportGen executes the SQL statement, and gets data from data source. First, it inserts some rows (11 rows per record) according to the number of records. Second, it copies the source range into the all added ranges. And then it adds page breaks per 4 records. Finally, it puts the data into the report by records.

Example 3, Variable Table Report with Group

The following function will makes the Word 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
;
```

Result

The variable table report defined in the report template:

	Customer Name	Contact Name	Phone/Fax	Address

The variable table report generated in the report:

	Customer Name	Contact Name	Phone/Fax	Address
A				
	Alfreds Futterkiste	Maria Anders	Phone: 030-0074321	Obere Str. 57
			Fax: 030-0076545	Berlin, Germany
				12209
	Ana Trujillo Emparedados y helados	Ana Trujillo	Phone: (5) 555-4729	Avda. de la Constitución 2222
			Fax: (5) 555-3745	México D.F., Mexico
				05021
B				
	Berglunds snabbköp	Christina Berglund	Phone: 0921-12 34 65	Berguvsvägen 8
			Fax: 0921-12 34 67	Luleå, Sweden
				S-958 22
	Blauer See Delikatessen	Hanna Moos	Phone: 0621-08460	Forsterstr. 57
			Fax: 0621-08924	Mannheim, Germany
				68306

Remarks

1. The SQL statement will get the information of customers including company name, contact name, phone, fax, address, city name, country name, and postal code.

2. The default type is “var”. It is a variable table report.

3. table=1. WDReportGen will put data into the first table in the report file.

4. group=1. WDReportGen will group data by the first letter of company name.

5. cell= A2,B3,C3,D3,D4,E3,E4,E5. These cells correspond to the first record.

6. copyrange=2:5. Because the default range is “B3:E5”, you must specify a range explicitly. WDReportGen will copy the range for each record.

7. There is no grouprange. WDReportGen will give a default value. The default grouprange is “2:5”.

8. pagebreak = 5r. WDReportGen will add a page break per 5 records.

9. WDReportGen executes the SQL statement, and gets data from data source. First, it groups the data. Second, it inserts some ranges according to the number of records. Next, it copies the source range into the all added ranges. And then it adds page breaks per 5 records. Finally, it puts the data into the report by records.

6.2 Form Report Function

The FORM REPORT function executes a SQL statement to get data from data source, and puts data into a range in the report file. You can put data from data source as text, list, title and table in the report file.

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.

WDReportGen 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 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 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 group. 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. WDReportGen 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 details and the area that includes all merge fields or quote fields for this group. For example, there are two groups, the range of group one contains all merge fields or quote fields for the group one and the range of group two, and the range of group two contains all merge fields or quote fields for the group two and the range of 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 or index number of field, but not simultaneously. In data source, you stored the path and file name of 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 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 WDReportGen 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 no page break.

- The **NODATA** argument specifies an option when no data are returned from data source. If the value is “delrange”, WDReportGen 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

The following function will makes the Word 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

;

Result

The form report defined in the report template:

«CategoryName»

«Description»

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

The form report generated in 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

Remarks

1. The SQL statement will get the information of products including product category, category description, product name, product ID, quantity per unit, unit price.

2. type=“form”. It is a form report.

3. cell=CategoryName, Description, ProductName, ProductID, QuantityPerUnit, UnitPrice. These merge fields or quote fields correspond to data source fields in the SQL statement.

4. range= Product. The bookmark “Product” defines the range for detail data. WDReportGen will copy the range for each record.

5. group=1,2. WDReportGen will group data by CategoryName and Description.

6. grouprange= Category. The bookmark “Category” defines the group range. WDReportGen will copy the range for each group.

7. WDReportGen executes the SQL statement, gets data from data source, and groups the data. It will process the report by record. First, it fetches a record. Second, it copies the source range/group range and inserts the range/group range into the report. Next, it puts the data into the report. And then it processes the next record.

6.3 Chart Function

The CHART function is used to make the charts.

Syntax

Chart(…)

sqlstatement

Arguments

TYPE = *reporttype*

CHART = *chart*

FILLORDER = *fillorder*

CELL= *celllist*

RANGE = *range*

GROUP= *grouplist*

GROUPRANGE = *grouprange*

RANGECOUNT = *rangecount*

NODATA = *nodataoption*

CONNECT = *datasource*

- The **TYPE** argument specifies the report type. Possible values are fix or var. “fix” means a fixed table report, and “var” means a variable table report. The default is var.
- 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 chart.
- The **FILLORDER** argument specifies the order in which WDReportGen fills data. Possible values are row or col. “row” means to fill data by rows. “col” means to fill data by columns. The 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. It identifies the cells in a datasheet or 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 …… WDReportGen 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 worksheet or datasheet to be used for the records. WDReportGen will skip or repeat the range for each record. You can reference a range of cells like “2:4” or “B2:D5”. The range “2:4” means 3 entire rows. The range “A:B” means 2 entire columns. The default range is the area that includes all cells for the records. For the fixed table report, WDReportGen will skip the range for each record. For the variable table report, it will insert the blank range for each record. The COPYRANGE argument is similar to the Range.

It will copy the original range to the range where data will be filled for each record. If there is no range or copyrange argument, the default for the fixed table report is range, and the default for the variable table report is copyrange.

- The **GROUP** argument specifies the group of 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 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 group in the worksheet. 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. WDReportGen will repeat the group range for each group. The range of group should contain the range of details and the area that includes all cells for this group. You reference a group range like "2:4" or "B2:D5". The ranges of groups must be same as the range of details.
- The **RANGECOUNT** argument specifies the number of blank range which you defined in the report template. It is valid when the type is "var". Possible values are 1 or 2. One means one blank range you defined, and two means two ranges. If you hope that the format of the last row/column border can be different from the others, you can define two blank ranges. Default is 1.
- The **NODATA** argument specifies an option when no data are returned from data source. It is valid when the type is "var". If the value is "delrange", WDReportGen 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.

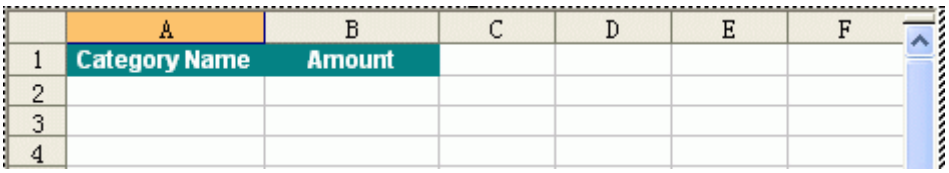
Example, Microsoft Excel Chart

The following function makes the chart: Sales by Categories.

```
@F3_2=CHART(chart=Chart3 cell=A2 rangecount=2)
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
```

Result

The worksheet of the chart defined in the report template:

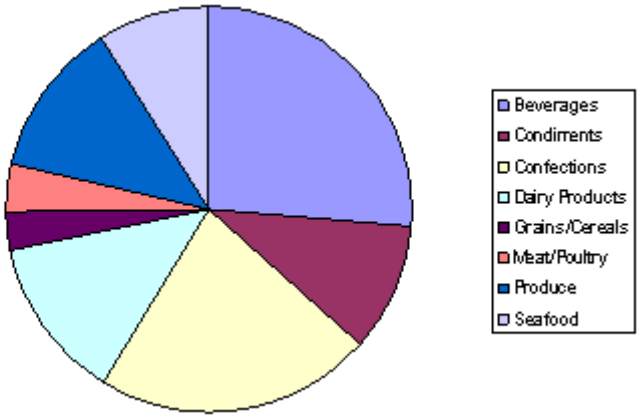


The chart defined in the report template is a blank chart.

The worksheet of the chart generated in the report:

	A	B	C	D	E	F
1	Category Name	Amount				
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:



Remarks

1. The SQL statement will get the information of sales by categories, including category name, and sales amount.

2. chart = Chart3. “Chart3” is the bookmark name of chart. It is an Excel chart. WDReportGen will put data into the worksheet of chart in the report file.

3. The default type is var. It is a variable table report.

4. The default fillorder is row. WDReportGen will fill data by rows.

5. cell=A2. The cells corresponding to the first record are “A2,B2”.

6. The default range is “A2:B2”. It is a copyrange.

7. rangecount=2. There are two blank ranges in the report template.

8. WDReportGen executes the SQL statement, and gets data from data source. First, it inserts some ranges according to the number of records.

Second, it copies the source range into the all added ranges. And then it puts the data into the report by records. Finally, it refreshes the chart.

6.4 ExecSQL Function

The EXECSQL function executes a SQL statement, but does not return result to 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

The following functions will create a table tmp0, and add some records into table. No result is returned to the report file.

```
@F1=EXECSQL()
CREATE TABLE tmp0 (
```

```
min_date DATE,
max_date DATE)
;
@F2=EXECSQL()
INSERT INTO tmp0
SELECT …
;
```

Chapter 7 Menus, Toolbar and Shortcut Keys

7.1 File Menu

The File menu offers the following commands:

New	Creates a new WRF file.
Open	Opens an existing WRF file.
Close	Closes an opened WRF file.
Save	Saves an opened WRF file using the same filename.
Save As	Saves an opened WRF file to a specified file name.
Open Report Template	Opens an existing report template file.
Open Report File	Opens an existing report file.
Open Log File	Opens an existing log file.
Recent Files	Opens last WRF files you closed.
Exit	Exits WDReportGen.

7.2 Edit Menu

The Edit menu offers the following commands:

Undo	Undo previous editing operation.
Redo	Redo an editing operation that you undid.
Cut	Deletes text from the document and moves it to the clipboard.
Copy	Copies text from the document to the clipboard.
Paste	Pastes text from the clipboard into the document.
Delete	Deletes the selection.
Select All	Selects the entire text.
Find	Finds the specified text.
Find Next	Finds the next matching text.
Replace	Replaces specific text with different text.
Go to	Goes to specified line or function in the document.

7.3 View Menu

The View menu offers the following commands:

Toolbar	Show or hide the toolbar.

Status Bar	Show or hide the status bar.

7.4 Report Menu

The Report menu offers the following commands:

Configuration	Configures the file names, data sources and parameters.
Run	Runs the WRF file to generate a report.

7.5 Tools Menu

The Tools menu offers the following commands:

Option	Sets options.

7.6 Help Menu

The Help menu offers the following commands:

Help Context	Starts the online help system.
Tutorial	Starts a brief step-by-step tutorial.
Tip of the Day	Displays a dialog containing a useful tip about WDReportGen.
Hints and Tips	Displays miscellaneous hints and tips on how to use WDReportGen productively.
Shortcut Keys	Shows the keyboard map.
Home Page	Takes you to the home page of WDReportGen web site.
Support	Takes you to the support page of WDReportGen web site.
Buy Now	Buy WDReportGen immediately.
About	Displays the version number of WDReportGen.

7.7 Toolbar

The toolbar provides quick access to many features. The buttons on the toolbar perform the following commands:

Buttons	Commands

	Creates a new WRF file.
	Opens an existing WRF file.
	Saves an opened WRF file using the same filename.
	Open the report template file.
	Open the report file.
	Deletes text from the document and moves it to the clipboard.
	Copies text from the document to the clipboard.
	Pastes text from the clipboard into the document.
	Undo previous editing operation.
	Redo an editing operation that you undid.
	Finds the specified text.
	Goes to specified line or function in the document.
	Configures the file names, data sources and parameters.
	Runs the WRF file to generate a report.
	Starts the online help system.
	Buy WDReportGen immediately.

7.8 Shortcut Keys

Shortcut Keys	Commands
Ctrl+N	Creates a new WRF file.
Ctrl+O	Opens an existing WRF file.
Ctrl+S	Saves an opened WRF file using the same filename.
Ctrl+Z	Undo previous editing operation.
Ctrl+Y	Redo an editing operation that you undid.
Ctrl+X	Deletes text from the document and moves it to the clipboard.
Ctrl+C	Copies text from the document to the clipboard.

Ctrl+V	Pastes text from the clipboard into the document.
Delete	Deletes the selection.
Ctrl+A	Selects the entire text.
Ctrl+F	Finds the specified text.
F3	Finds the next matching text.
Ctrl+H	Replaces specific text with different text.
Ctrl+G	Goes to specified line or function in the document.
F2	Configures the file names, data sources and parameters.
F5	Runs the WRF file to generate a report.
F1	Starts the online help system.

Chapter 8 Hints and Tips

You can run WDReportGen from the command line. The format is:

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

For example:

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

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

WDReportGen comes with a sample database Sample.mdb and some sample reports. You can use them when learning the program. To use the sample reports, 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. WDReportGen will get the text of 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 “\@”.

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.

You can create a chart using Microsoft Excel. To work with charts created in Excel, you must have Excel installed.

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.

You can protect the generated report so that it cannot be modified. To protect the report, select the **Protect Report** check box in the **Configuration** dialog box.

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

If you associate WDReportGen with the file extension “.wrf”, a WRF file with

the extension ".wrf" will open in WDReportGen when you double-click the file. The information:

File Extension: .wrf

Action: open

Application: "C:\Program Files\LJZsoft\wordreport.exe" "%1"

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 WRF file.

In the SQL statements, you can use parameters. To use parameters, you must define them first.

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

You should be careful to define a unique name for each parameter, because WDReportGen 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".

In the text editor window, 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.

You can use hints in the SQL statement for Oracle database.

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

To group data in a report, you should use GROUP argument in the report function.

In REPORT function, 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.

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

You can create reports with pictures using WDReportGen. You should store the path and name of 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 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.

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 WDReportGen to generate report. The two steps can be written into a batch file.

WDReportGen5 cannot refresh fields include table of contents and (Formula) fields.

To refresh table of contents and (Formula) fields, you can use WDReportGen4. First, use WDReportGen5 to generate a temporary document. And then use WDReportGen4 to refresh table of contents and (Formula) fields.

Chapter 9 WRF File Reference

9.1 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 WRF file. “Version 2.0” is the version of 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 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.

9.2 [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, WDReportGen 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, WDReportGen 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, WDReportGen 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, WDReportGen 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 data sources. If the value is Y, the passwords will be saved in an encrypted format. If the value is N, the passwords will be saved in plain text.

9.3 [FILE] Section

[FILE] section contains information about files.

ReportTemplateFileName=<templatefilename>

This setting specifies the name of report template file. <templatefilename> value is the name and path of 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 WRF file.

ReportFileName=<reportfilename>

This setting specifies the name of report file. <reportfilename> value is the name and path of 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 WRF file. In <reportfilename>, you can use parameters.

ProtectReport=Y/N

This setting specifies whether the report generated is protected. If the value is Y, the report is protected, and cannot 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.

LogFileName=<logfilename>

This setting specifies the name of log file. <logfilename> value is the name and path of 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 WRF file. In <logfilename>, you can use parameters.

9.4 [PARAMETER] Section

[PARAMETER] section contains information about parameters.

Name1=<name1>

Name2=<name2>

……

Name10=<name10>

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

Chapter 10 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. WDReportGen gets the text from the cell, and outputs the result using it as the format expression. In fact, WDReportGen calls the format function in Visual Basic. For more information about format, refer to Format Function in *Visual Basic for Applications Reference*.

10.1 Formats for Numeric Values

10.1.1 Different Formats for Different Numeric Values

A user-defined format expression for numbers can have from one to three sections separated by semicolons. If the Style argument of the Format function contains one of the predefined 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 applies to negative values.
Three sections	The first section applies to positive values, the second applies to negative values, and the third applies to zeros.

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”

10.1.2 Predefined Numeric Formats

The following table identifies the predefined numeric format names. These may be used by name as the style argument:

Format name	Description
General Number, G, or g	Displays number with no thousand separator.
Currency, C, or c	Displays number with thousand separator, if appropriate; displays two digits to the right of the decimal separator. Output is based on system locale settings.
Fixed, F, or f	Displays at least one digit to the left and two digits to the right of the decimal separator.
Standard, N, or n	Displays number with thousand separator, at least one digit to the left and two digits to the right of the decimal separator.
Percent	Displays number multiplied by 100 with a percent sign (%) appended immediately to the right; always displays two digits to the right of the decimal separator.
P, or p	Displays number with thousandths separator multiplied by 100 with a percent sign (%) appended to the right and separated by a single space; always displays two digits to the right of the decimal separator.
Scientific	Uses standard scientific notation, providing two significant digits.
E, or e	Uses standard scientific notation, providing six significant digits.

D, or d	Displays number as a string that contains the value of the number in Decimal (base 10) format. This option is supported for integral types (Byte, Short, Integer, Long) only.
X, or x	Displays number as a string that contains the value of the number in Hexadecimal (base 16) format. This option is supported for integral types (Byte, Short, Integer, Long) only.
Yes/No	Displays No if number is 0; otherwise, displays Yes.
True/False	Displays False if number is 0; otherwise, displays True.
On/Off	Displays Off if number is 0; otherwise, displays On.

10.1.3 User-Defined Numeric Formats

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. Displays a digit or a zero. If the expression has a digit in the position where the zero appears in the format string, display it; otherwise, displays 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, displays 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, rounds 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, displays the extra digits without modification.
(#)	Digit placeholder. Displays a digit or nothing. If the expression has a digit in the position where the # character appears in the format string, displays it; otherwise, displays 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 fewer digits than there are # characters on either side of the decimal separator in the format expression.
(.)	Decimal placeholder. The decimal placeholder determines how many digits are displayed to the left and right of the decimal separator. If the format expression contains only # characters to the left of this symbol; numbers smaller than 1 begin with a decimal separator. To display a leading zero displayed with fractional numbers, use zero as the first digit placeholder to the left of the decimal separator. In some locales, a comma is used as the decimal separator. The actual character used as a decimal placeholder in the formatted output depends on the number format recognized by your system. Thus, you should use the period as the decimal placeholder in your formats even if you are in a locale that uses a comma as a decimal placeholder. The formatted string will appear in the format correct for the locale.
(%)	Percent placeholder. Multiplies the expression by 100. The percent character (%) is inserted in the position where it

	appears in the format string.
(,)	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 #). A thousand separator immediately to the left of the decimal separator (whether or not a decimal is specified) or as the rightmost character in the string means “scale the number by dividing it by 1,000, rounding as needed.” Numbers smaller than 1,000 but greater or equal to 500 are displayed as 1, and numbers smaller than 500 are displayed as 0. Two adjacent thousand separators in this position scale by a factor of 1 million, and an additional factor of 1,000 for each additional separator. Multiple separators in any position other than immediately to the left of the decimal separator or the rightmost position in the string are treated simply as specifying the use of a thousand separator. In some locales, a period is used as a thousand separator. The actual character used as the thousand separator in the formatted output depends on the Number Format recognized by your system. Thus, you should use the comma as the thousand separator in your formats even if you are in a locale that uses a period as a thousand separator. The formatted string will appear in the format correct for the locale. For example, consider the three following format strings: “#,0.”, which uses the thousands separator to format the

	number 100 million as the string “100,000,000”. “#0,.”, which uses scaling by a factor of one thousand to format the number 100 million as the string “100000”. “#,0,.”, which uses the thousands separator and scaling by one thousand to format the number 100 million as the string “100,000”.
(:)	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.
(E- E+ e- e+)	Scientific format. If the format expression contains at least one digit placeholder (0 or #) to the left 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 left 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. You must also include digit placeholders to the right of this symbol to get correct formatting.
- + $ ( )	Literal characters. These characters are displayed exactly as typed in the format string. To display a character other than one of those listed, precede it with a backslash (\) or enclose

	it in double quotation marks (“ “).
(\)	Displays 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”)	Displays the string inside the double quotation marks (“ “). To include a string in the style argument from within code, you must use Chr(34) to enclose the text (34 is the character code for a quotation mark (“)).

10.2 Formats for Date/Time Values

10.2.1 Predefined Date/Time Formats

The following table identifies the predefined date and time format names. These may be used by name as the style argument:

Format Name	Description
General Date, or G	Displays a date and/or time. For example, 3/12/2008 11:07:31 AM. Date display is determined by your

	application's current culture value.
Long Date, Medium Date, or D	Displays a date according to your current culture's long date format. For example, Wednesday, March 12, 2008.
Short Date, or d	Displays a date using your current culture's short date format. For example, 3/12/2008.
Long Time, Medium Time, or T	Displays a time using your current culture's long time format; typically includes hours, minutes, seconds. For example, 11:07:31 AM.
Short Time or t	Displays a time using your current culture's short time format. For example, 11:07 AM.
f	Displays the long date and short time according to your current culture's format. For example, Wednesday, March 12, 2008 11:07 AM.
F	Displays the long date and long time according to your current culture's format. For example, Wednesday, March 12, 2008 11:07:31 AM.
g	Displays the short date and short time according to your current culture's format. For example, 3/12/2008 11:07 AM.
M, m	Displays the month and the day of a date. For example, March 12.
R, r	Formats the date according to the RFC1123Pattern property. For example, Wed, 12 Mar 2008 11:07:31 GMT. The formatted date does not adjust the value of the date and time. You must adjust the Date/Time value to GMT before calling the Format function.
s	Formats the date and time as a sortable index. For

	example, 2008-03-12T11:07:31.
u	Formats the date and time as a GMT sortable index. For example, 2008-03-12 11:07:31Z.
U	Formats the date and time with the long date and long time as GMT. For example, Wednesday, March 12, 2008 6:07:31 PM.
Y, y	Formats the date as the year and month. For example, March, 2008.

10.2.2 User-Defined Date/Time Formats

The following table shows characters you can use to create user-defined date/time formats. Unlike in earlier versions of Visual Basic, these format characters are case-sensitive.

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 that is used as the time separator in formatted output is determined by your application's current culture value.
(/)	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 that is used as the date separator in formatted output is determined by your application's current culture.
(%)	Used to indicate that the following character should be read as a single-letter format without regard to any trailing letters. Also

	used to indicate that a single-letter format is read as a user-defined format. See what follows for additional details.
d	Displays the day as a number without a leading zero (for example, 1).
dd	Displays the day as a number with a leading zero (for example, 01).
ddd	Displays the day as an abbreviation (for example, Sun).
dddd	Displays the day as a full name (for example, Sunday).
M	Displays the month as a number without a leading zero (for example, January is represented as 1). Use %M if this is the only character in your user-defined numeric format.
MM	Displays the month as a number with a leading zero (for example, 01/12/01).
MMM	Displays the month as an abbreviation (for example, Jan).
MMMM	Displays the month as a full month name (for example, January).
gg	Displays the period/era string (for example, A.D.).
h	Displays the hour as a number without leading zeros using the 12-hour clock (for example, 1:15:15 PM).
hh	Displays the hour as a number with leading zeros using the 12-hour clock (for example, 01:15:15 PM).
H	Displays the hour as a number without leading zeros using the 24-hour clock (for example, 1:15:15).
HH	Displays the hour as a number with leading zeros using the 24-hour clock (for example, 01:15:15).
m	Displays the minute as a number without leading zeros (for example, 12:1:15).
mm	Displays the minute as a number with leading zeros (for

	example, 12:01:15).
s	Displays the second as a number without leading zeros (for example, 12:15:5).
ss	Displays the second as a number with leading zeros (for example, 12:15:05).
f	Displays fractions of seconds. For example ff displays hundredths of seconds, whereas ffff displays ten-thousandths of seconds. You may use up to seven f symbols in your user-defined format.
t	Uses the 12-hour clock and displays an uppercase A for any hour before noon; displays an uppercase P for any hour between noon and 11:59 P.M.
tt	For locales that use a 12-hour clock, displays an uppercase AM with any hour before noon; displays an uppercase PM with any hour between noon and 11:59 P.M. For locales that use a 24-hour clock, displays nothing.
y	Displays the year number (0-9) without leading zeros.
yy	Displays the year in two-digit numeric format with a leading zero, if applicable.
yyy	Displays the year in four-digit numeric format.
yyyy	Displays the year in four-digit numeric format.
z	Displays the timezone offset without a leading zero (for example, -8).
zz	Displays the timezone offset with a leading zero (for example, -08).
zzz	Displays the full timezone offset (for example, -08:00).

Chapter 11 License and Support

11.1 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 WDReportGen indicates your acceptance of this agreement (“License”).

NO WARRANTY

WDREPORTGEN 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

WDReportGen 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.

Use

Each Licensed Copy of this Software may either be used by a user who uses the software personally on one or more computers, or installed on a single computer used nonsimultaneously by multiple people, but not both. This is not a concurrent use license. If you install the Software onto a multi-user platform, server or network, each and every individual user of the Software must be licensed separately.

Distribution

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.
You may not modify, copy, share, distribute, re-sell, transfer or sub-license the full version of this software except that you may make copies for archive purposes only. Any such unauthorized use shall result in immediate and automatic termination of this license.

Edition

There are 2 types of editions issued for WDReportGen.
1) WDReportGen Standard Edition
User can execute no more than 50 SQL statements in one report processing.

2) WDReportGen Professional Edition
User can execute SQL statements unlimitedly in one report processing.

11.2 Technical Support

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

- Go to our support web site at:
http://www.ljzsoft.com/support.htm
- Send email to support@ljzsoft.com