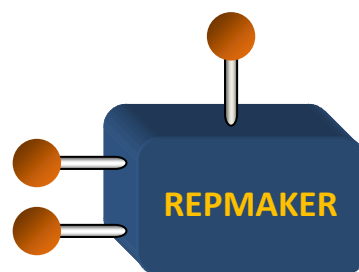


ООО “ЮНИСОФТ”



REPMAKER.PVT v.1.1

Генератор сводных таблиц и диаграмм для
MS Excel

Руководство по настройке и эксплуатации

2009 год

Оглавление

Введение	4
Используемые сокращения	4
Что такое “REPMAKER.PVT” ?	4
О сводных таблицах	5
Основные возможности	5
Дополнительные источники информации	6
Общие сведения	7
Процесс генерации	7
Пример отчета	7
Структура файла конфигурации	8
Секции и ключи	8
[OUTPUT]	8
[PIVOTTABLE]	9
[PARAMETERS]	10
[SQL]	11
COM – интерфейс	12
Свойства (properties)	12
Методы (methods)	13
Мастер сводных таблиц и диаграмм	21
Шаг 1. Выбор источника данных	22
Шаг 2. Выбор столбцов	22
Шаг 3. Указание условий отбора	23
Шаг 4. Указание порядка сортировки	23
Шаг 5. Предварительное макетирование	24
Шаг 6. Указание характеристик сводной таблицы	24

Шаг 7. Последние инструкции	24
Приложение 1. Образец файла конфигурации	26
Приложение 2. Примеры использования COM-объекта.....	27
PowerBuilder	27
Visual Basic	27



Введение

Данное руководство ориентировано на программистов, имеющих опыт написания программ для операционных систем Windows 98/Me/NT/2000/2003/XP, а также знакомых с технологиями COM/ActiveX и OLE-automation.

Используемые сокращения

В этом документе встречаются следующие сокращения:

- | | |
|--------|------------------------------|
| ✓ COM | Component Object Model |
| ✓ HTML | Hyper Text Markup Language |
| ✓ MTS | Microsoft Transaction Server |
| ✓ OLE | Object Linking and Embedding |
| ✓ SQL | Structured Query Language |
| ✓ URL | Universal Resource Location |

Что такое “REPMaker.PVT” ?

REPMaker.PVT – это внешний COM/OLE сервер, предназначенный для динамического создания (генерации) сводной таблицы или диаграммы в программе Microsoft Excel® на основе настраиваемого файла конфигурации. REPMaker.PVT позволяет также запускать встроенный мастер создания сводных таблиц, который может быть достаточно легко интегрирован в любую программную среду, поддерживающую технологию OLE Automation:

-  Powersoft PowerBuilder
-  Microsoft Visual Basic
-  Microsoft Visual C++
-  Microsoft Visual FoxPro
-  Microsoft C#
-  Borland Delphi
-  Borland C++ Builder
-  Приложения семейства Microsoft Office (Excel, Word, Access и др.)
-  И другие



О сводных таблицах

Рисунок 1. Пример сводной таблицы

Регион	Восток	Товар	Молоко	Мясо	Общий итог
Сумма продаж	Май	Батулин	22 977р.	17 578р.	40 555р.
Май	Белов	22 977р.	17 578р.	40 555р.	40 555р.
Май всего	Батулин	10 017р.	7 711р.	17 728р.	17 728р.
Июнь	Белов	6 805р.	5 575р.	12 380р.	12 380р.
Июнь всего	Батулин	16 823р.	13 286р.	30 109р.	30 109р.
Общий итог	Белов	39 799р.	30 864р.	70 663р.	70 663р.

Сводная таблица представляет собой отчет логически и конструктивно состоящий из четырех разделов: 1) Строки, 2) Колонки, 3) Данные, 4) Фильтр. При этом информация во всех разделах может представлять собой многоуровневую иерархическую структуру.

Например, на Рисунке 1 данные по строке представляют двухуровневую структуру: Месяц-Продавец.

Удобство при работе со сводной таблицей в MS Excel заключается не только в способности многомерного отображения данных и итоговых значений, но и в возможности динамически ее перестраивать путем перетаскивания колонок между любыми разделами таблицы. Таким образом, представление и уровень детализации анализа может настраиваться конечным пользователем самостоятельно. Все настройки сохраняются при закрытии и восстанавливаются при повторном открытии файла. При этом пользователь имеет возможность сделать автоматическое обновление данных в таблице для отображения последних изменений, сделанных в базе данных.

Основные возможности

Поддержка баз данных

Широкий диапазон источников данных основан на применении интегрированного с Excel приложения Microsoft Query®, установка которого необходима для правильной работы REPMaker.PVT. Этот модуль применяется в MS Excel как промежуточное звено доступа к любому внешнему источнику, для которого на компьютере установлен ODBC-драйвер, при этом, каждый отдельный отчет может быть создан на основе собственного соединения.

Мощный набор функций

Разработчик может использовать во всех выражениях, применяемых внутри файла конфигурации, как богатые возможности встроенного языка, так и большое количество дополнительных функций по работе с числами, строками, базами данных и пр. (Более подробно см. Документ *Описание процедур и функций. Руководство программиста*).

Параметризованные отчеты

В файле конфигурации, при необходимости, можно описать неограниченное количество параметров отчета с необязательными значениями по умолчанию. Это позволяет создавать универсальные конфигурации, присваивая на этапе исполнения значения только необходимым параметрам.

Простота настройки

В связи с тем, что настройка и описание макета отчета хранятся в файле конфигурации стандартного формата (ini-файл), редактирование его возможно с использованием любого текстового процессора.

Поддержка технологии COM+/MTS

Компонент **REPMAKER.PVT** может быть установлен как “Приложение COM+” в программе “Служба компонентов” или как объект MTS сервера, что позволяет провести его более тонкую настройку и администрирование, настроить права доступа на основе ролей, а также обеспечить к нему удаленный доступ через DCOM (более подробно см. “Руководство по установке REPMAKER.PVT”). Средства авторизации на основе ролей могут быть использованы не только для целей ограничения доступа к объекту на различных уровнях (вплоть до свойств и методов), но также и для управления доступом к макету отчета в целом.

Дополнительные источники информации

Перечень и синтаксис доступных для использования процедур и функций описан в документе *REPMAKER.Процедуры и функции. Руководство по программированию.*



Структура файла конфигурации

Файл конфигурации представляет собой обычный текстовый ini-файл, информация в котором структурирована по *секциям* и *ключам*. Ключ – это элемент данных, записанный в виде пары: Код=Значение. Секция – это группа ключей, имеющая собственное имя и представленная в следующем виде: [Имя_секции]. Строки, начинающиеся с “;” (точка с запятой) являются комментариями. Полный пример описания файла конфигурации приведен в *Приложении 1* данного руководства.

Секции и ключи

[OUTPUT]

Данная секция ключей предназначена для более детального описания макета создаваемой сводной таблицы. При отсутствии в файле этой секции, верхний левый угол отчета сводной таблицы располагается в ячейке, указанной в ключе RANGE секции PIVOTTABLE.

Ключи

В этой секции могут находиться строки, каждая из которых представляет собой сложносоставной ключ с описанием команды: в какой ячейке, или диапазоне ячеек (RANGE) вывести заданное значение (VALUE) с указанными атрибутами (BOLD, ITALIC, UNDERLINE, SIZE). Разделителем ключей является символ “;”. Пример, такой строки приведен ниже:

Пример

RANGE=A2; VALUE=@Name; BOLD=1; SIZE=12

Имя	Тип	Описание
RANGE	string	Скалярное значение, обозначающее имя ячейки электронной таблицы
VALUE	string	Значение, которое будет присвоено ячейке. Может описываться следующим образом: 1. Скалярное значение 2. Выражение, начинающееся с символа ‘&’, результат выполнения которого присваивается ячейке 3. Зарезервированное слово: @Name – Название сводной таблицы, определенное в ключе <i>Name</i> секции PIVOTTABLE @PivotTable – собственно сама сводная таблица
BOLD	int	(exp) Выражение, в результате вычисления которого, значению ячейки присваивается атрибут полужирного начертания (1) или нет (0)
UNDERLINE	int	(exp) Выражение, в результате вычисления которого, значению ячейки присваивается атрибут <u>подчеркивания</u> (1) или нет (0)
ITALIC	int	(exp) Выражение, в результате вычисления которого, значению ячейки присваивается атрибут <i>курсивного</i> начертания (1) или нет (0)

SIZE int (exp) Размер шрифта

[PIVOTTABLE]

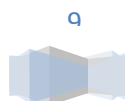
Эта секция описывает непосредственно сам макет сводной таблицы, а также атрибуты, относящиеся как к таблице, так и к листу книги Excel® в целом.

Ключи

Имя ключа	Описание
<i>Name</i>	String Наименование сводной таблицы
<i>Range</i>	String Адрес (имя) ячейки, являющейся верхним левым углом сводной таблицы
<i>PageSetup.Orientation</i>	Ориентация листа при печати: Landscape, Portrait
<i>Format</i>	Int Формат сводной таблицы. Стандартное значение должно быть в диапазоне 0..21. По умолчанию 0. Значение формата 99 – является расширением стандартных форматов и обозначает необходимость выделения итогов по строкам и столбцам полужирным шрифтом.
<i>RowGrand</i>	Int Отображать итоги по строкам: 1 (да), 0 (нет)
<i>ColumnGrand</i>	Int Отображать итоги по столбцам: 1 (да), 0 (нет)
<i>RowFields</i>	Список полей через запятую, которые должны быть выведены в виде строк
<i>ColumnFields</i>	Список полей через запятую, которые должны быть выведены в виде столбцов
<i>PageFields</i>	Список полей через запятую, которые должны быть выведены в разделе фильтра (страницы)
<i>PivotFields</i>	Список полей через запятую, которые должны быть выведены в виде значений суммирования (числа) или подсчета количества (не числа)
<i>Zoom</i>	Процент масштабирования. По умолчанию 100%.

Каждое поле, входящее в список ключей *RowFields*, *ColumnFields*, *PageFields* может иметь атрибуты, значения которых вводятся внутри полукруглых скобок “()” после имени поля в формате пары: Атрибут=Значение. Символом-разделителем описания атрибутов является “;” (точка с запятой). Атрибуты полей, и их описание приводятся в следующей таблице:

Имя атрибута	Описание
<i>AutoFit</i>	Автоматическая установка ширины колонки
<i>Caption</i>	Текст заголовка



Format Формат вывода

Subtotals Индексы функций расчета итогов через “,” (запятую).

Индекс	Значение
1	Automatic
2	Sum
3	Count
4	Average
5	Max
6	Min
7	Product
8	Count Nums
9	StdDev
10	StdDevp
11	Var
12	Varp

Пример

```
; Строки
RowFields=ИмяГруппы(SubTotals=1),Название,Код,ЕИ
; Колонки
ColumnFields=Дата (Format=ДД/ММ)
; Страницы
PageFields=ИмяГрЗак
; Вычисления
PivotFields=Кол-во (Format=0.###)
```

[PARAMETERS]

Секция с описанием параметров отчета относится к категории служебных. Каждая строка данного раздела представляет собой описание одного параметра отчета, состоящее из трех атрибутов: имя параметра (ID), тип данных (DATATYPE), значение по умолчанию (DEFAULT). Тип данных может принимать одно из следующих значений: string, number, date, datetime, time. Если тип данных не указан явно, используется неявное значение string. Значение, присвоенное параметру, может быть использовано во всех разделах, в которых, так или иначе, происходит вычисление выражений путем простого указания его имени. В секции [SQL] ссылка на значение параметра осуществляется по схеме:

:<имя_параметра>.

Пример

```
[PARAMETERS]
; Отчету передается диапазон дат, по умолчанию – текущий месяц, ID сотрудника
ID=pBegDate; DATATYPE=date;                      DEFAULT=(DateBOM(Today()))
ID=pEndDate; DATATYPE=date;                      DEFAULT=(DateEOM(Today()))
ID=pEmpID;    DATATYPE=number;                      DEFAULT=NULL
```

[SQL]

Внутреннее содержание секции SQL состоит из текста SQL-выражения (SELECT..FROM..), или вызова хранимой процедуры (execute [owner.]proc(p1,p2..pN)), возвращающей набор данных. Внутри текста SQL-оператора допустимы ссылки на значения параметров отчета. В последнем случае, перед именем параметра ставится символ ":" (двоеточие).

В тексте оператора допустимы также макровыражения на языке программирования PowerScript, которые ограничиваются с обеих сторон символами "<<" и ">>". Результатом выполнения такого макровыражения всегда является строка, которая заменяет его описание в исходном тексте.

После выполнения SQL-предложения, на имена колонок, присутствующих в операторе SELECT, можно ссылаться при описании ключей *RowFields*, *ColumnFields*, *PageFields*, *PivotFields* в секции PIVOTTABLE. Если вместо списка колонок, используется символ "*", имена колонок будут созданы автоматически на основании физической структуры таблиц(ы) базы данных.

Если СУБД не поддерживает локализованных имен колонок, можно использовать специальный синтаксис, в котором после имени колонки или выражения, в фигурных скобках "{ }", необходимо указать локализованное название колонки (выражения). Это же имя, при необходимости, должно присутствовать в списке полей, указанных выше ключей секции PIVOTTABLE.

Пример**[SQL]**

```
select P_CODE as {Код},
       P_NAME as {Название},
       DUEDATE as {Дата},
       ROUND(QTY,5) as {Кол-во},
       U_CODE as {ЕИ},
       G_CODE as {КодГруппы},
       G_NAME as {ИмяГруппы},
       C_CODE as {Поставщик},
       M_CODE as {КодГрЗак},
       M_NAME as {ИмяГрЗак},
       SUMV as {Сумма}
from MRPPLO
where FIRM_ID = :pFirmID and
      TYPEOF = :pTypeOf
```



COM – интерфейс

Сервисы, предоставляемые генератором сводных таблиц и диаграмм REPMaker.PVT, могут быть вызваны из любого языка программирования, поддерживающего OLE – automation. Для этого, необходимо в первую очередь создать экземпляр самого ActiveX-компонента - сервера услуг, используя программный идентификатор (**REPMaker.PVT**) или идентификатор класса CLASSID. Конкретный способ (метод), при помощи которого создается и инициализируется ActiveX-компонент, зависит от используемого языка программирования. После того, как объект успешно создан и ссылка на него присвоена некоторой глобальной переменной программы-контейнера, можно выполнять методы объекта и присваивать/читать значения его свойств, используя следующий синтаксис: <oVar>.Method() или <oVar>.Property. Примеры программного кода, написанные на различных языках можно посмотреть в *Приложении 2* данного руководства, а также в директории \SAMPLES, созданной после инсталляции REPMaker.

Ниже перечислены основные свойства и методы COM-сервера **RepMaker.PVT**.

Свойства (properties)

Свойства общего назначения

DstPath	String. Полный путь до директории, в которой будут создаваться результирующие отчеты. Только чтение.
ErrCode	Long. Содержит числовой код результата выполнения последней операции над объектом. Только чтение.
ErrText	String. Содержит текстовое описание результата выполнения последней операции над объектом. Только чтение.
HostName	String. Имя компьютера, на котором выполнен запуск приложения. Только чтение.
SrcPath	String. Полный путь до директории с файлами-шаблонами отчетов. Только чтение.

Параметры соединения с базой данных

DBMS	String. Тип источника данных: ODBC,OLE DB, Microsoft SQL Server, JDB JDBC и др. Обязательный атрибут.
Database	String. Имя базы данных
DBParm	String. Параметры подключения к базе данных в формате: <атрибут>=<значение>;...
DBPass	String. Пароль для входа в базу данных
Lock	String. Уровень изоляции
LogID	String. Имя пользователя для входа в сервер базы данных
LogPass	String. Пароль для входа в сервер базы данных
ServerName	String. Имя сервера базы данных
UserID	String. Имя пользователя для входа в базу данных



Методы (methods)

CheckRole()	Проверка пользователя на принадлежность к роли COM+/MTS сервера
DbConInit()	Инициализация параметров подключения на основе данных, расположенных в ini-файле или системном реестре
DbConnect()	Подключение к источнику данных
DbDisconnect()	Отсоединение от источника данных
Generate()	Создание сводной таблицы (диаграммы), на основе макета, загруженного с помощью функции Load(), или вызов Мастера создания
IsConnected()	Проверка на наличие соединения с базой данных
Load()	Загрузка файла конфигурации с разбором (parsing) его составляющих
SetDstPath()	Установка полного пути до директории, в которой будут создаваться сгенерированные отчеты
SetParam()	Установка значения параметра по его имени
SetParams()	Установка значений набора параметров
SetSrcPath()	Установка полного пути до директории, в которой находятся файлы-шаблоны

Передача параметров отчету

Имена, типы и значения по умолчанию параметров отчета описываются в специальной секции файла конфигурации отчета: PARAMETERS. Фактические значения параметров могут быть установлены во время выполнения программы одним из следующих способов:

- 1) Выполнение метода объекта SetParam() для установки значения одного параметра по его имени
- 2) Выполнение метода объекта SetParams() для установки значений сразу нескольких параметров
- 3) Указание исходного или результирующего файла в методах типа Load(), MakeFile(), Generate() с дополнительной передачей значений параметров по их имени, например:

```
oRMK.Load("c://temp//res0001.pvt?pEmpID=" + 120 + "&pDate='10.01.2003'")
```

- 4) Указание исходного или результирующего файла в методах типа Load(), MakeFile(), Generate() с дополнительной передачей значений параметров по их порядковому номеру, например:

```
oRMK.Load("c://temp//res0001.pvt ? #1=" + 120 + "&#2='10.01.2003' & #3=NULL" )
```



Далее идет более подробное описание методов COM-объекта:

DbConInit(filename: String, section: String): Integer

Назначение

Инициализация параметров подключения на основе данных, расположенных в ini-файле или системном реестре

Аргументы

Filename	String. Полный путь до файла с описанием параметров подключения к источнику данных. Если значение переданного параметра равно \$registry , аргументу <i>section</i> должно быть присвоено значение ключа системного реестра ОС Windows.
Section	String. Имя секции ini-файла или ключа системного реестра, в котором находится описание параметров подключения в формате: <атрибут>=<значение>

Возвращаемое значение

Integer. При успешном завершении операции метод возвращает значение 1, иначе -1.

Описание

Метод считывает параметры подключения к источнику данных, описанных в указанном ini-файле или системном реестре Windows, в разделе, указанном в параметре *section*. Значения параметров подключения присваиваются соответствующим одноименным свойствам COM-объекта.

Примеры

```
oRepMaker1.DbConInit("myfile.ini", "Connection")
oRepMaker1.DbConnect()
...
oRepMaker2.DbConInit("$registry", "HKEY_CURRENT_USER\Software\MyFirm\MyProduct")
oRepMaker2.DbConnect()
...
```

```
myfile.ini
-----
[Connection]
DBMS=ODBC
DBParm=ConnectString='DSN=MyDSN; UID=MyUserName'
; RU (Read Uncommitted)
Lock=RU
```

Метод

DbConnect(): Integer



Назначение

Подключение к источнику данных на основе описанных параметров

Аргументы

Нет

Возвращаемое значение

Integer. Результат попытки подключения. Успешное подключение возвращает значение 0. При возникновении ошибки в процессе подключения, метод возвращает значение меньше 0, а свойству *ErrText* присваивается текст описания ошибки.

Описание

Данный метод осуществляет подключение объекта к источнику данных, описанного при помощи свойств *DBMS*, *DbParm*, *Database* и др. Такое подключение является для объекта глобальным и используется как значение по умолчанию в тех макетах отчетов, у которых в служебном тэге SQL отсутствует ссылка на внутренний источник данных (DBS).

Примеры

```
oRepMaker.DBMS="ODBC"  
oRepMaker.DBParm="ConnectionString='DSN=TestDB; UID=guest'; SQLCache=10"  
oRepMaker.DbConnect()  
...
```

Метод

DbDisconnect() : Integer

Назначение

Отключение от источника данных

Аргументы

Нет

Возвращаемое значение

Integer. Результат попытки разрыва соединения. Успешное отключение возвращает значение 0. При возникновении ошибки в процессе отключения, метод возвращает значение меньше 0, а свойству *ErrText* присваивается текст описания ошибки.



Описание

Данный метод осуществляет отключение объекта от источника данных, к которому было установлено последнее соединение.

Примеры

```
oRepMaker.dbDisonnect()
```

Метод

Generate(typeof: Integer): Integer

Назначение

Непосредственная генерация сводной таблицы или диаграммы в программе Microsoft Excel®

Аргументы

typeof Integer[in]. Вид генерируемого отчета. Может принимать следующие значения:

- 0 – Генерация сводной таблицы на основе загруженных настроек методом Load()
- 1 – Генерация диаграммы на основе загруженных настроек методом Load()
- 2 – Генерация сводной таблицы или диаграммы с использованием Мастера

Возвращаемое значение

Integer. При успешном выполнении загрузки шаблона, метод возвращает 1. Иначе -1.

Описание

Метод предназначен для генерации результирующего отчета сводной таблицы или диаграммы в программе Microsoft Excel® с указанием типа (режима) его формирования. При значении параметра *typeof* равным 2, метод автоматически вызывает Мастера, который формирует отчеты на основе файлов конфигураций, находящихся в директории, заданной методом SetSrcPath(). Результат предыдущего вызова метода Load() при этом игнорируется.

Примеры

```
Constant int PVT_MASTER = 2
Integer li_rc

li_rc = oRM.Generate(PVT_MASTER)
```



Метод

Load(filename:String) : Integer

Назначение

Загрузка файла конфигурации с разбором (parsing) его составляющих

Аргументы

filename	String [in]. Путь до файла конфигурации. Если переданное значение аргумента содержит только имя файла, полный путь до файла определяется с учетом свойства объекта SrcPath.
----------	---

Возвращаемое значение

Integer. При успешном выполнении загрузки шаблона, метод возвращает 1. Иначе -1.

Описание

Данный метод объекта читает указанный в параметре *filename* файл шаблона отчета и осуществляет на основе его описания загрузку внутренних структур. Непосредственное формирование выходного документа отчета осуществляется методом Generate().

Примеры

```
// PowerBuilder
if oRM.Load("c:\templ\t001.pvt") = 1 then
    oRM.SetParams({"01.07.2003","10.07.2003"})
    // генерируем сводную таблицу...
    li_rc = oRM.Generate(0)
end if
```

Метод

SetDstPath(path: String) : Integer

Назначение

Установка пути по умолчанию до директории с результирующими отчетами

Аргументы

Path	String. Полный путь до директории
------	-----------------------------------

Возвращаемое значение

Integer. При успешном выполнении возвращается 1, иначе (-1).



Описание

Функция устанавливает значение свойства DstPath, которое используется функцией MakeFile() в случае, если в переданном ей в качестве параметра файле отсутствует описание пути. При этом в описании пути может присутствовать ссылка на переменную памяти в формате %VAR_NAME%.

Примеры

```
oRMK = CreateObject("REPMAKER.PVT")  
oRMK.SetDstPath("%TEMP%\HTMLREP")
```

Метод

SetParam(name:String, value:String): Integer

Назначение

Установка значения параметра по его имени

Аргументы

name	String [in]. Имя параметра
value	String [in]. Значение параметра

Возвращаемое значение

Integer. Метод возвращает 1 в случае успешного присвоения значения указанному параметру. В противном случае возвращается -1.

Описание

Данный метод применяется для присвоения значения параметру с именем *name*, который описан в файле шаблона отчета, загруженного в объект последним путем вызова метода Load(). При наличии у параметра значения по умолчанию, это значение переписывается указанным значением *value*.

Примеры

```
// PowerBuilder  
string ls_filename = "departs.pvt"  
if oRMK.Load("d:\templates\"+ls_filename) = 1 then  
    oRMK.SetParam("pFirmID", 1)  
    if oRMK.Generate(0) <> 1 then  
        MessageBox("Error No "+string(oRMK.ResultCode), oRMK.ResultText, Exclamation!)  
    end if  
else  
    MessageBox("Error No "+string(oRMK.ResultCode), oRMK.ResultText, Exclamation!)  
end if
```



Метод

SetParams(arr[]: String): Integer

Назначение

Установка значений набора параметров

Аргументы

arr [in] Строковый массив. Значения передаваемых параметров

Возвращаемое значение

Integer. При успешном выполнении возвращается 1, иначе -1.

Описание

Функция выполняет присвоение значений сразу нескольким параметрам отчета. При этом порядковый номер элемента массива *arr* соответствует порядковому номеру параметра в файле шаблоне. Если количество элементов массива меньше, чем количество параметров, оставшимся параметрам присваивается значение по умолчанию. Если количество элементов больше, чем количество параметров, лишние элементы массива *arr* игнорируются.

Примеры

```
// PowerBuilder
string ls_filename = "depstats.pvt"
if oRMK.Load("d:\templates\"+ls_filename) = 1 then
    // Сформировать статистику работы подразделений организации [1], за первый квартал 2004г.
    oRMK.SetParams( {1, "01.01.2004", "31.03.2004"})
    if oRMK.Generate(0) <> 1 then
        MessageBox("Error No "+string(oRMK.ResultCode), oRMK.ResultText, Exclamation!)
    end if
else
    MessageBox("Error No "+string(oRMK.ResultCode), oRMK.ResultText, Exclamation!)
end if
```

Метод

SetSrcPath(path: String): Integer

Назначение

Установка пути по умолчанию до директории, в которой хранятся макеты отчетов

Аргументы

Path String. Полный путь до директории



Возвращаемое значение

Integer. При успешном выполнении возвращается 1, иначе (-1).

Описание

Функция устанавливает значение свойства SrcPath, которое используется функцией Load() в случае, если в переданном ей в качестве параметра файле отсутствует описание пути. При этом в описании пути может присутствовать ссылка на переменную памяти в формате %VAR_NAME%.

Примеры

```
oRMK = CreateObject("REPMAKER.PVT")  
oRMK.SetSrcPath("C:\REPSRV\TEMPLATES")
```



Мастер сводных таблиц и диаграмм

Мастер предназначен для пошагового определения параметров и макета формируемого отчета типа сводная таблица или сводная диаграмма в программе Microsoft Excel®. Как уже упоминалось выше, такие типы отчетов очень удобны при проведении всестороннего анализа имеющихся данных, выявления трендов, расчета итоговых сумм в различной детализации, а также представления информации в графическом виде.

Мастер позволяет построить отчет несколькими способами:

1. "С нуля" в режиме конструктора
2. На основе ответов, сохраненных при предыдущем выполнении мастера
3. На основе ранее созданного при помощи мастера отчета (шаблона) с указанием только параметров отбора данных

Вариант 1

Процедура создания нового отчета в режиме конструктора сводится к следующим шагам:

1. Выбрать необходимый источник описания данных из предлагаемого настраиваемого списка.
2. Выбрать из списка элементы данных, которые будут использоваться при построении аналитического куба.
3. Определить необходимые условия отбора данных.
4. Указать порядок сортировки данных.
5. Описать предварительно макет будущего отчета.
6. Указать необходимые параметры визуального отображения.
7. Дать команду на формирование отчета с возможностью предварительного сохранения всех настроек.

Вариант 2

Процедуру формирования отчета можно значительно ускорить, если воспользоваться файлами с предварительно сохраненными его характеристиками. Для чего, на первом этапе щелкните на вкладке *Открыть* и выберите файл конфигурации с расширением .UPVT, воспользовавшись кнопкой *Найти*. Дальнейшие шаги связаны только с уточнением параметров выбора данных и самой формы отчета.

Вариант 3

Самый быстрый способ генерации на основе ранее созданной и визуально настроенной формы отчета - шаблона. Шаблон создается и визуально настраивается после первого выполнения мастера. При последующем запуске мастера, на самом первом его этапе, необходимо щелкнуть на вкладке *Открыть* и выбрать шаблон с расширением .XLS или .XLSX (MS Office 2007), воспользовавшись кнопкой *Найти*. Дальнейшие шаги связаны только с уточнением параметров выборки данных. Этот вариант особенно удобен тем, что конечный пользователь может самостоятельно создавать и многократно использовать собственные шаблоны отчетов, не прибегая к помощи разработчика.

Ниже приводится подробное описание каждого шага выполнения Мастера. На любом шаге можно как вернуться назад к предыдущему этапу, так и отказаться совсем от услуг Мастера с аннулированием всей введенной информации.

Шаг 1. Выбор источника данных

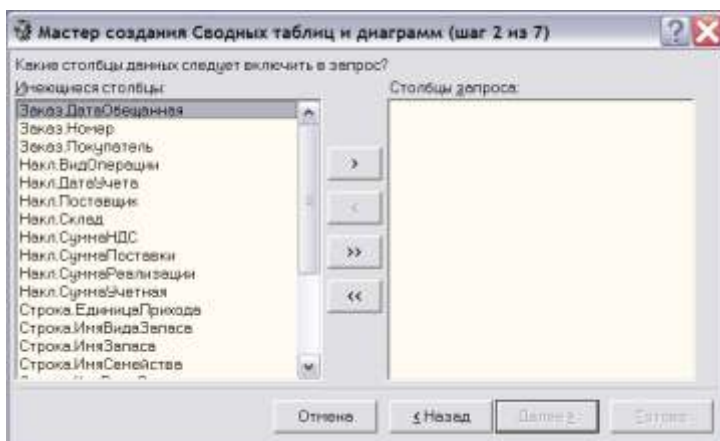
На начальном этапе работы мастера требуется указать источник информации для генерации отчета. В режиме конструктора (вкладка *Новый*) укажите источник в поле *Источник данных* и выберите вид отчета – сводная таблица или диаграмма.

Чтобы открыть предварительно созданный шаблон (XLS) или конфигурационный файл (UPVT), щелкните на вкладке *Открыть*, а затем на кнопке *Найти*.



Шаг 2. Выбор столбцов

На этом шаге необходимо указать Мастеру колонки, данные которых будут задействованы в дальнейшем при генерации отчета. В окне имеются два списка с наименованиями колонок: слева – полный перечень, справа – выбранные для отчета. Перемещение колонок между списками осуществляется щелчком на кнопках одиночного (>,<) и массового (>>,<<) действия.

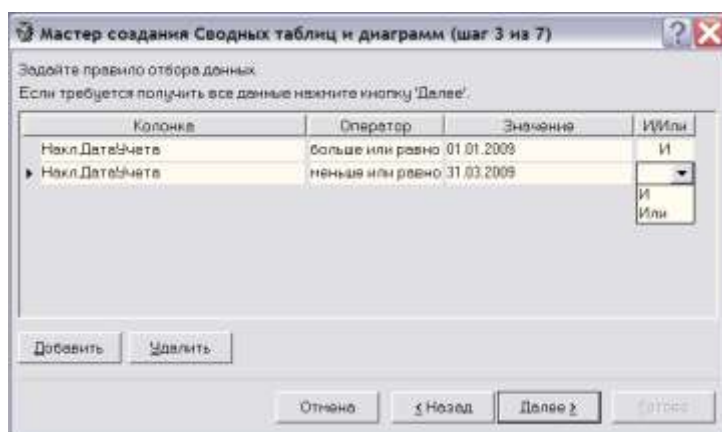


Шаг 3. Указание условий отбора

На этом этапе требуется указать Мастеру условия для выборки информации из источника данных. Каждое условие описывается одной строкой отображаемой таблицы. Элементами условия являются:

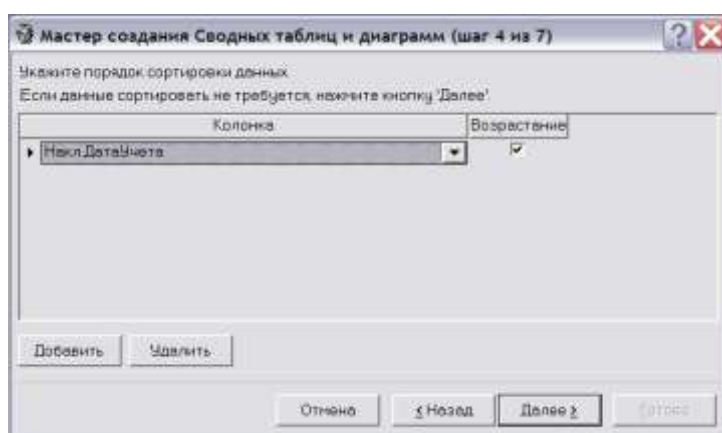
1. *Колонка* – наименование колонки, выбираемое из списка, сформированного на предыдущем этапе
2. *Оператор* – выражение, применяемое для сравнения (выбирается из списка)
3. *Значение* – вводимое пользователем значение, применяемое для сравнения с имеющимися в колонке данными

Несколько условий могут соединяться между собой словами И / ИЛИ.



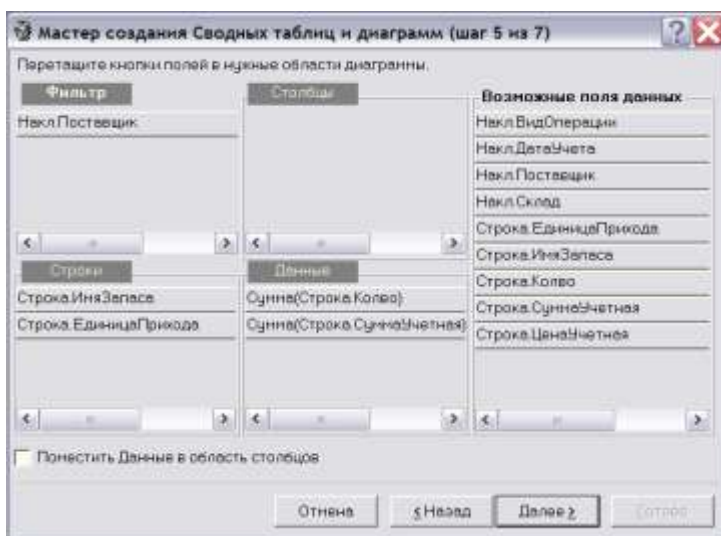
Шаг 4. Указание порядка сортировки

При необходимости можно указать колонки и направление сортировки информации .



Шаг 5. Предварительное макетирование

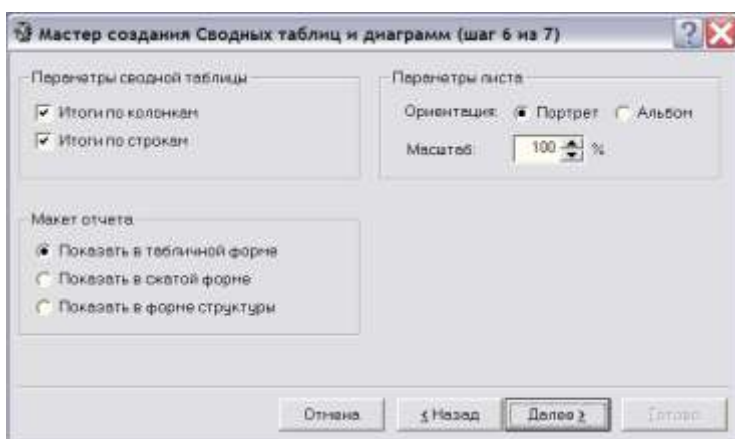
Как уже описывалось выше, отчет сводной таблицы состоит из четырех разделов. На этом шаге пользователю предлагается перетащить при помощи мыши колонки из общего списка в список, соответствующий одному из разделов сводной таблицы.



Если в списке *Данные* находится более одной строки, можно воспользоваться переключателем *Поместить Данные в область столбцов*. Это приведет к построению сводной таблицы с данными, расположенными горизонтально.

Шаг 6. Указание характеристик сводной таблицы

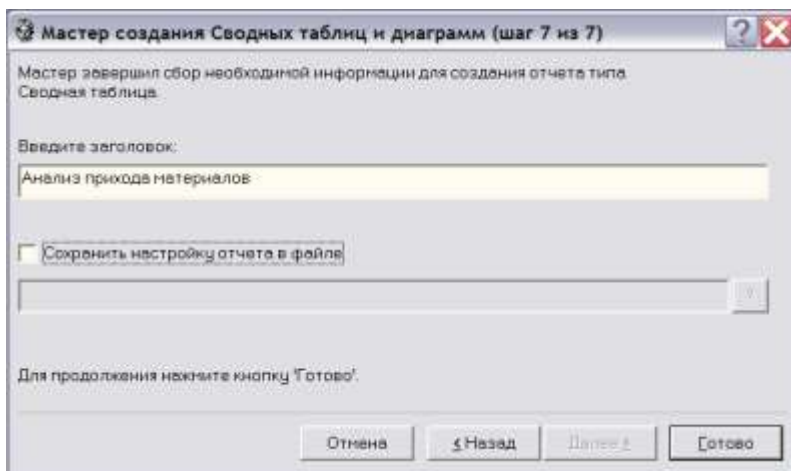
Этот шаг позволяет указать Мастеру визуальные характеристики создаваемого отчета.



Шаг 7. Последние инструкции

На заключительном шаге необходимо указать текст заголовка сводной таблицы, и (при необходимости) файл конфигурации, в котором будут сохранена вся информация, собранная в процессе работы Мастера. В дальнейшем, для генерации отчета с аналогичными характеристиками, достаточно выбрать

этот файл на начальном этапе после запуска Мастера. Для запуска процесса генерации отчета необходимо щелкнуть на кнопке *Готово*.



Приложение 1. Образец файла конфигурации

```
;Имя файла: sample.pvt
[OUTPUT]
RANGE=A1; VALUE=&"Предприятие: "+dlookup$("name","firm","id="+pFirmID)
RANGE=A2; VALUE=@Name; BOLD=1; SIZE=12
RANGE=A6; VALUE=@PivotTable

[PIVOTTABLE]
; Название таблицы
Name=План потребности материалов (план заявок на закупку)
; Размещение в Excel
Range=A3
PageSetup.Orientation=Landscape
Format= 99
RowGrand=1
ColumnGrand=1
; Строки
RowFields=ИмяГруппы(SubTotals=1),Название,Код,ЕИ
; Колонки
ColumnFields=Дата(Format=ДД/ММ)
; Страницы
PageFields=ИмяГрпЗак
; Вычисления
PivotFields=Кол-во(Format=0.###)

[PARAMETERS]
ID=pFirmID; DATATYPE=number; DEFAULT=@FIRM_ID
ID=pTypeOf; DATATYPE=string; DEFAULT='B'

[SQL]
select P_CODE as {Код},
       P_NAME as {Название},
       DUEDATE as {Дата},
       ROUND(QTY,5) as {Кол-во},
       U_CODE as {ЕИ},
       G_CODE as {КодГруппы},
       G_NAME as {ИмяГруппы},
       C_CODE as {Поставщик},
       M_CODE as {КодГрпЗак},
       M_NAME as {ИмяГрпЗак},
       SUMMA as {Сумма}
from MRPPLO
where FIRM_ID = :pFirmID and
      TYPEOF = :pTypeOf
```



Приложение 2. Примеры использования COM-объекта

PowerBuilder

```
OleObject oRMK
integer rc
try
    // создание и инициализация COM-объекта (COM-object creation and initialization)
    oRMK = CREATE OleObject
    rc = oRMK.ConnectToNewObject("RepMaker.PVT")
    if rc < 0 then
        MessageBox("Connection Error", "Error number "+string(rc))
        Return
    end if
    // Параметры соединения с БД (database connection string)
    oRMK.DBMS = "ODBC"
    oRMK.DBParm = "ConnectionString='DSN=TestDB;UID=Empl'"
    oRMK.DbConnect()
    // Загрузка HTML-шаблона (loading HTML-template)
    rc = oRMK.Load("c:\templates\templ1.pvt")
    // Установка значения параметра (set parameter's value)
    rc = oRMK.SetParam("FirmID", 5)
    // Генерация отчета сводной таблицы
    oRMK.Generate(0)
catch (runtimeerror e)
    MessageBox("error", e.Text)
finally
    oRMK.DisconnectObject( )
end try
Return
```

Visual Basic

```
Sub Main()
Dim s As String
Dim i As Integer
Dim oRMK As Object
Dim res As Double
Dim pars(1) As String

pars(0) = "01.01.2005"
pars(1) = "31.03.2005"
Set oRMK = CreateObject("RepMaker.PVT")
If Not IsObject(oRMK) Then Return
res = oRMK.Load("c:\templates\templ01.pvt")
res = oRMK.SetParams(pars)
oRMK.DBMS = "ODBC"
oRMK.DBParm = "ConnectionString='DSN=TestDB'"
oRMK.DbConnect()
' Генерация диаграммы
oRMK.Generate 1
End Sub
```

