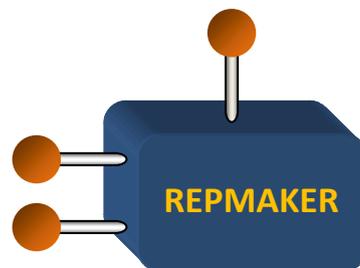


ООО “ЮНИСОФТ”



РЕРМАКЕР.HTML v.1.2

Генератор отчетов в формате WEB-страниц

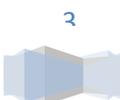
Руководство по программированию

2009 год

Оглавление

Введение	4
Используемые сокращения	4
Что такое “REPMAKER.HTML” ?	4
Основные возможности	4
Дополнительные источники информации	6
Принцип формирования отчетов	7
Процесс генерации	7
Протокол “rep://”	7
Пример отчета.....	8
Структура макета	9
Основные разделы	9
Служебные разделы.....	10
Дескрипторы разметки	11
\$\$_HEADER.....	12
\$\$_PAGEHEADER.....	13
\$\$_GROUPHEAD#<N>	13
\$\$_DETAIL.....	14
\$\$_GROUPFOOT#<N>	15
\$\$_PAGEFOOTER.....	15
\$\$_FOOTER.....	16
\$\$_INIT	16
\$\$_INFO.....	17
\$\$_PARAMETERS.....	17
\$\$_VARS	18
\$\$_DBCONN	19

\$\$_SQL.....	20
\$\$_DONE	21
\$\$_INCLUDE.....	22
Файл конфигурации.....	23
COM – интерфейс.....	24
Свойства (properties)	24
Методы (methods)	25
Приложение 1. Образец шаблона.....	37
Приложение 2. Примеры использования COM-объекта.....	39
PowerBuilder	39
Visual Basic	39
JavaScript.....	39



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.HTML” ?

RepMaker.HTML – это внешний COM/OLE сервер, предназначенный для динамического создания (генерации) отчетов в формате WEB – страниц на основе полученной информации из различных источников или баз данных. Он может быть использован в любой программной среде, поддерживающей технологию OLE Automation:

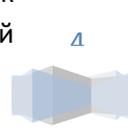
- ✚ Powersoft PowerBuilder
- ✚ Microsoft Visual Basic
- ✚ Microsoft Visual C++
- ✚ Microsoft Visual FoxPro
- ✚ Microsoft C#
- ✚ Borland Delphi
- ✚ Borland C++ Builder
- ✚ Приложения семейства Microsoft Office (Excel, Word, Access и др.)
- ✚ И другие

Основные возможности

Поддержка баз данных

REPMAKER.HTML может работать с информацией, полученной из различных баз данных, обращаясь к ним по следующим интерфейсам: ODBC, JDBC, OLE DB, ADO .NET, Native DB drivers. При этом каждый отдельный отчет может быть создан на основе собственного соединения.

Визуальное проектирование



Привязка данных к полям отчета осуществляется на основе элемента *Bookmark* языка HTML, что позволяет использовать практически любой существующий визуальный редактор HTML – страниц для проектирования макетов (шаблонов) отчетов в режиме WYS-I-WYG. С появлением на рынке бесплатного профессионального продукта Microsoft Web Developer Express Edition, входящего в состав Visual Studio 2008, кроме богатых возможностей визуального проектирования, разработчику стали доступны средства создания и тестирования SQL-запросов с помощью встроенного построителя. После отладки и контроля полученных данных, результирующая строка, содержащая оператор SELECT или вызов хранимой процедуры, может быть встроена в макет отчета.

Вложенные отчеты

REPMAKER.HTML позволяет создавать вложенные друг в друга отчеты, каждый из которых может самостоятельно соединяться с нужной ему базой данных и осуществлять необходимые выборки. Уровень вложений ограничен только ресурсами системы. В итоге создается сложно - составной документ, который выглядит для конечного пользователя как одно целое. Связь между вложенными отчетами осуществляется на уровне параметров и глобальных переменных.

Стили отчетов

Текущая версия REPMAKER.HTML поддерживает только ленточные band-ориентированные отчеты с возможностью неограниченного количества группировок.

Мощный набор функций

Разработчик может использовать во всех выражениях, выполняемых внутри отчета, как богатые возможности встроенного языка, так и большое количество дополнительных функций по работе с числами, строками, базами данных и пр.

Параметризованные отчеты

Отчеты, генерируемые REPMAKER.HTML являются параметризованными. Кроме этого, параметры отчета могут иметь значения по умолчанию, что позволяет проектировать универсальные шаблоны, присваивая им в момент генерации значения только необходимых параметров.

Макросы в SQL – предложениях

В предложение SELECT, на базе которого формируется основной набор данных для отчета, может быть вставлено неограниченное количество макросов, или блоков кода, результат выполнения которых становится неотъемлемой частью исходного предложения. Это позволяет формировать сложные запросы к базе данных “на лету” в зависимости от внешних условий.

Поддержка кодировок OEM/ANSI

Данные, используемые для создания отчетов, могут храниться в базе данных в различной кодировке. При указании используемой кодировки, REPMAKER автоматически осуществляет преобразование данных для их правильного отображения в результирующем документе.

Простота поддержки макета

В связи с тем, что шаблон отчета – это обычный текстовый файл, содержащий описание страницы в формате HTML, редактирование его возможно с использованием любого текстового процессора. При хорошем знании языка HTML, файл шаблона может быть создан в таком редакторе практически с “нуля”.

Поддержка технологии COM+/MTS

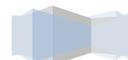
Компонент **REPMAKER.HTML** может быть установлен как “Приложение COM+” в программе “Служба компонентов” или как объект MTS сервера, что позволяет провести его более тонкую настройку и администрирование, настроить права доступа на основе ролей, а также обеспечить к нему удаленный доступ через DCOM (более подробно см. “Руководство по установке REPMAKER.HTML”). 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.

Экспорт отчета в Microsoft Word® и Excel®

При установленных программах Word и Excel, входящих в пакет Microsoft Office®, **REPMAKER.HTML** позволяет преобразовать исходный HTML-документ в соответствующий формат с максимальным приближением полученной формы к оригиналу.

Дополнительные источники информации

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



Принцип формирования отчетов

Процедура формирования готового отчета есть процесс наполнения предварительно созданного макета, или шаблона, документа данными из одной или нескольких таблиц базы данных. Как уже упоминалось выше, для того, чтобы создать шаблон отчета, можно воспользоваться любым графическим редактором HTML-страниц. На первом этапе в нем рисуется вся статическая часть, или бланк документа, с использованием, при необходимости, таких элементов HTML, как static text, table, picture и др. Далее, в нужные места созданного макета вставляются именованные метки (Bookmarks), вместо которых во время исполнения будут подставлены соответствующие данные. И, наконец, в текст полученного HTML-файла встраиваются специальные служебные метки с обозначением логических частей, или зон конечного документа. Последний этап требует от дизайнера хотя бы минимальных знаний самого языка разметки HTML.

В общем, под статической частью понимается текст HTML-документа (в том числе и конструкции языков DHTML, XML, JavaScript, VBScript, CSS и др.), который никаким образом не обрабатывается, а просто переносится в итоговый документ без всяких изменений. РЕPMAKER.HTML анализирует синтаксис и семантику и затем соответствующим образом транслирует только определенные участки текста шаблона. В зависимости от местоположения текста относительно той или иной логической области (band), он может быть многократно размножен в итоговом документе.

Процесс генерации

Ниже приведена примерная схема процесса генерации результирующего отчета на основе шаблона (явный вызов):

1. запустить программу
2. создать объект РЕPMAKER.HTML (способ создания зависит от языка программирования)
3. установить параметры соединения с БД (при необходимости)
4. установить связь с источником данных (при необходимости)
5. установить пути по умолчанию (при необходимости)
6. загрузить файл макета
7. получить результат в нужном виде (html-файл, html-текст и др.)

Средством просмотра может являться любой доступный интернет-обозреватель (Internet Explorer, Netscape и др.). Для просмотра html-страниц непосредственно из программы можно использовать элемент ACTIVEХ *Microsoft WebBrowser Control*, устанавливающийся на компьютере при инсталляции IE (рекомендуется версия 5.5 и выше). Примеры программного кода, написанные на различных языках программирования, можно посмотреть Приложении 2 данного руководства, а также в директории \SAMPLES, созданной после инсталляции РЕPMAKER.

Протокол "гер://"

Специальный протокол гер://, указанный в адресе URL какой-либо гиперссылки (), говорит серверу РЕPMAKER, что следующая за протоколом строка символов является путем до файла шаблона. Адреса URL с таким протоколом обрабатываются специальным образом функцией

CheckNavigateBefore(), которая должна быть вызвана в событии BeforeNavigate2 (Navigating) элемента WebBrowser.

Пример отчета

На следующем рисунке приведен пример отчета, созданного объектом REPMAKER.HTML:

Предприятие: ООО "ВИКОНТ"
Склад: СМ-10.1 Склад сырья и материалов

Ведомость поступления
за период с 01.04.2006 по 30.04.2006

Ном. номер	Наименование товара, работ, услуг	Количество		По приходу		По учету	
		приход	осв. ЕН	Цена	Сумма	Цена	Сумма
БИНОТЕК							
	Дабко DC 5804	0,8 тн	0,8тн	198 203,39	158 562,71	198 203,39	158 562,71
	Полнол 8	1,376 тн	1,376тн	91 915,25	126 475,39	91 915,25	126 475,39
	Итого по поставщику:	2,176	2,176		285 038,10		285 038,10
В.А.ДИДУР ООО ИВН							
	Полнол	2,1 тн	2,1тн	55 000,00	115 500,00	55 000,00	115 500,00
	Полнол	0,42 тн	0,42тн	69 000,00	28 980,00	69 000,00	28 980,00
	Итого по поставщику:	2,52	2,52		144 480,00		144 480,00
ВХС							
	Запон 5003-2-15	1 тн	1тн	44 491,53	44 491,53	44 491,53	44 491,53
	Итого по поставщику:	1	1		44 491,53		44 491,53
ВХТ							
	Воранол 360	14,2 тн	14,2тн	62 300,00	884 660,00	62 300,00	884 660,00
	Воранол 450	1,8 тн	1,8тн	57 323,00	103 181,40	57 323,00	103 181,40
	Дабко DC 197	1 тн	1тн	198 226,98	198 226,98	198 226,98	198 226,98
	Дисеперформанс	0,005 тн	0,005тн	50 000,00	250,00	50 000,00	250,00
	Омронат CR 30-20	19 тн	19тн	60 549,52	150 440,88	60 549,52	150 440,88
	Супрасек 5005	20 тн	20тн	78 900,00	578 000,00	78 900,00	578 000,00
	ТДМ	1,25 тн	1,25тн	74 576,27	93 220,34	74 576,27	93 220,34
	Триклоротицефат	20 тн	20тн	49 584,00	991 680,00	49 584,00	991 680,00
	Триптоламин тели м.б.	0,1 тн	0,1тн	34 100,00	3 410,00	34 100,00	3 410,00
	Итого по поставщику:	77,355	77,355		5 003 069,60		5 003 069,60
ВЫМШЕ.И.Д							
	Корчинат ПМ м. 45-25	4 тн	4тн	57 305,08	229 220,32	57 305,08	229 220,32

Готово | ЕСТРАОМН | СМ-10.1

Рисунок 1. Пример отчета, созданного REPMAKER.HTML

Структура макета

Как было описано в предыдущей главе, кроме статической части, макет документа содержит специальные метки для обозначения логических разделов отчета. При этом разделы делятся на две основные категории: основные и служебные.

Основные разделы

RepMaker.HTML поддерживает описание следующих основных разделов для вывода информации (bands):

Код	Название	Описание
Header	Заголовок отчета	Печатается один раз на самом верху документа
Page header	Заголовок страницы	Печатается в начале каждой страницы
Group header	Заголовок группы	Печатается в заголовке каждой группы данных основного источника
Detail	Данные основного источника	Печатается для каждой записи основного источника данных
Group footer	Подножие группы	Печатается в подножии каждой группы данных основного источника
Page footer	Подножие страницы	Печатается в конце каждой страницы
Footer	Подножие отчета	Печатается один раз в самом низу документа

*Наличие каждого раздела в тексте шаблона не является строго обязательным. Однако, раздел **HEADER** рекомендуется все-таки включать, при отсутствии других разделов в макете.*

Ниже приведено максимальное количество разделов, которое анализируется при трансляции текста шаблона:

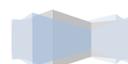
Header	1
Page header	1
Group header	не ограничено
Detail	1
Group footer	равно фактическому количеству Group header
Page footer	1
Footer	1

Служебные разделы

Помимо основных областей, непосредственно связанных с отображаемой (печатаемой) частью документа, существуют специальные, или служебные разделы. Два таких раздела - SQL и VARS, являются обязательными при условии, что отчет непосредственно связан с информацией из базы данных. В следующей таблице перечислены служебные разделы и их назначение:

Код	Название	Описание
INIT	Инициализация	Команды, выполняемые при старте отчета
INFO	Настройки	Описание настроек отчета
PARAMETERS	Параметры отчета	Блок описания параметров отчета
DBCONN	Источник данных	Описание параметров подключения к Базе данных
SQL	SQL-оператор	Блок описания оператора SQL и способа подключения к источнику данных
VARS	Переменные	Подставляемые данные
INCLUDE	Включения	Вставка содержимого другого файла
DONE	Завершение	Команды, выполняемые при закрытии отчета

Все служебные разделы, кроме INCLUDE, должны быть описаны в макете отчета только один раз. Как в основных, так и в служебных разделах, при необходимости вычисления какого-либо выражения, могут быть использованы различные функции, описанные далее.



Дескрипторы разметки

Как упоминалось выше, содержание текста макета HTML-документа анализируется сервером RepMaker.HTML определенным образом. При этом, анализу подвергаются только ключевые теги, или дескрипторы, реализованные в виде HTML-комментариев вида `<!--$$_ ... -->`, остальной текст копируется в выходной файл без изменения. В следующей таблице перечислены коды и назначение каждой ключевой метки:

Дескриптор	Описание
<i>Основные разделы</i>	
\$\$_HEADER	Начало заголовка отчета
\$\$_PAGEHEADER	Начало заголовка страницы
\$\$_GROUPHEAD#<N>	Начало заголовка группы с порядковым номером N
\$\$_DETAIL	Начало области вывода данных основного источника
\$\$_GROUPFOOT#<N>	Начало подножия группы с порядковым номером N
\$\$_PAGEFOOTER	Начало подножия страницы
\$\$_FOOTER	Начало подножия отчета
<i>Служебные разделы</i>	
\$\$_INIT	Блок команд инициализации
\$\$_DONE	Блок команд завершения
\$\$_INFO	Описание дополнительных настроек
\$\$_PARAMETERS	Описание параметров отчета
\$\$_DBS	Описание источника данных
\$\$_SQL	SQL-выражение, возвращающее набор данных
\$\$_VARS	Привязка элементов данных к соответствующим меткам шаблона
\$\$_INCLUDE	Команда на включение в шаблон текста из другого файла перед началом общего процесса обработки

Порядок описания, а также местонахождение в тексте шаблона любого служебного раздела не имеет значения. Однако, придерживаясь структуры самого HTML-документа, рекомендуется описывать

разделы служебной группы в содержании тега <HEAD>...</HEAD>. Ниже приведена схема описания служебного раздела:

```
<!--$$_XXXXXXX [атрибуты]
... содержание ...
-->
```

Из схемы видно, что описание раздела состоит из некоторого внутреннего содержания (content), располагающегося между метками начала и окончания раздела, а также необязательным набором атрибутов (attributes), разделенных символом ";" (точка с запятой). Каждый атрибут представляет собой описание пары: Имя=Значение. При этом формат описания содержательной части дескриптора зависит от имени и функции служебного раздела.

Ниже приведен короткий пример описания параметров, используемых набором данных отчета:

```
<!--$$_PARAMETERS
  ID=pDStart;  DATATYPE=date;  DEFAULT=01.03.2000
  ID=pDEnd;    DATATYPE=date;  DEFAULT=31.03.2000
-->
```

В отличие от служебных, порядок следования основных разделов в макете существенно влияет на формирование результирующего документа. При этом их порядок, в основном, диктуется смысловым содержанием документа и практически совпадает с последовательностью описания основных разделов из предыдущей таблицы. Другим отличием является то, что дескрипторы основных разделов имеют не внутреннее, а внешнее содержание, расположенное сразу за описанием тега и оканчивающееся началом следующего тега из этой - же группы или концом файла. Основные разделы также могут иметь атрибуты в формате пары: Имя=Значение.

Далее каждый дескриптор разметки описывается более подробно:

\$\$_HEADER

Назначение:

Начало описания раздела заголовка отчета.

Атрибуты:

VISIBLE (exp) Выражение, возвращающее числовой результат и определяющее условие видимости всего раздела в итоговом документе. Раздел не отображается, если результат вычисления выражения равен 0. Необязательный атрибут. По умолчанию 1.

Описание

Этот тег рекомендуется размещать сразу после тега <BODY>. Раздел HEADER формируется один раз в самой верхней части отчета.

Пример

```
<BODY>  
  
<!--$$_HEADER-->  
  
<table>...</table>
```

\$\$_PAGEHEADER

Назначение:

Начало описания раздела заголовка страницы

Атрибуты:

VISIBLE (exp) Выражение, возвращающее числовой результат и определяющее условие видимости всего раздела в итоговом документе. Раздел не отображается, если результат вычисления выражения равен 0. Необязательный атрибут. По умолчанию 1.

Описание

Этот тег рекомендуется размещать сразу после дескриптора <TABLE> (если есть). Раздел PAGEHEADER автоматически формируется для каждой печатной страницы.

Пример

```
<table>  
<!--$$_PAGEHEADER-->  
<tr>  
<td>...</td>  
<td>...</td>  
<td>...</td>  
</tr>  
<!--$$_DETAIL-->  
...
```

\$\$_GROUPHEAD#<N>

Назначение:

Начало описания раздела заголовка группы/подгруппы с порядковым номером, или уровнем <N>.

Атрибуты:

VISIBLE (exp) Выражение, возвращающее числовой результат и определяющее условие видимости всего раздела в итоговом документе. Раздел не отображается, если результат вычисления выражения равен 0. Необязательный атрибут. По умолчанию 1.

GROUPBY (exp) Выражение, результат которого служит основанием для вычисления условия группировки данных. Формат описания сложного выражения: <exp1>|<exp2>|...|<expN>. Выражение может быть как простым, в виде элемента основного набора данных (колонки), заданного в разделе \$\$_SQL, так и более сложным с применением функций и

условных операторов. Обязательный атрибут.

- SORTBY** (exp) Выражение, применяемое для сортировки самой группы. Чаще всего, в качестве выражения выступают статистические функции типа sum(), avg() и пр. Необязательный атрибут.
- PAGEBREAK** (exp) Выражение, возвращающее числовой результат и управляющее режимом, когда каждое значение группы печатается с нового листа. Переход на новый лист осуществляется, если значение выражения равно 1. В остальных случаях новый лист не создается. Необязательный атрибут. По умолчанию 0.

Описание

Основанием для появления в отчете нового раздела, соответствующего N-му уровню группировки, является смена значения выражения, заданного в атрибуте GROUPBY. Для правильной работы механизма группировки, выражение GROUPBY должно соответствовать порядку сортировки, заданному для набора данных в разделе \$\$_SQL. Значение выражения атрибута SORTBY является критерием, на основании которого выполняется последующая сортировка данных внутри текущей группы. Порядковые номера групп должны начинаться с 1, и не могут иметь пропусков значений.

Пример

```
<!--$$_GROUPHEAD#1 GROUPBY=dep_id|status; SORTBY=sum(salary)-->
```

\$\$_DETAIL

Назначение

Начало раздела, который транслируется и копируется в итоговый документ для каждой записи основного набора данных.

Атрибуты

- VISIBLE** (exp) Выражение, возвращающее числовой результат и определяющее условие видимости всего раздела в итоговом документе. Раздел не отображается, если результат вычисления выражения равен 0. Необязательный атрибут. По умолчанию 1.

Описание

Данный дескриптор обозначает начало раздела, который после обработки должен быть добавлен в результирующий документ для каждой записи основного набора данных при значении выражения VISIBLE, равном 1.

Пример

```
<!--$$_DETAIL VISIBLE=if(Status='A', 1, 0)-->  
<tr>  
<td>...</td>  
<td>...</td>  
</tr>
```

\$\$_GROUPFOOT#<N>

Назначение

Начало раздела подножия группы/подгруппы с порядковым номером, или уровнем <N>.

Атрибуты

VISIBLE (exp) Выражение, возвращающее числовой результат и определяющее условие видимости всего раздела в итоговом документе. Раздел не отображается, если результат вычисления выражения равен 0. Необязательный атрибут. По умолчанию 1.

Описание

Данный тег служит для указания того, что следующий за ним текст HTML является содержимым подножия группы с уровнем (порядковым номером) <N>. При этом, необходимым условием появления этого раздела является присутствие в более верхней части шаблона соответствующего тега \$\$_GROUPHEAD.

Пример

```
<!--$_GROUPFOOT#1-->
```

\$\$_PAGEFOOTER

Назначение:

Начало описания раздела подножия страницы

Атрибуты:

VISIBLE (exp) Выражение, возвращающее числовой результат и определяющее условие видимости всего раздела в итоговом документе. Раздел не отображается, если результат вычисления выражения равен 0. Необязательный атрибут. По умолчанию 1.

Описание

Этот тег рекомендуется размещать перед дескриптором \$\$_FOOTER (если есть). Раздел PAGEFOOTER формируется автоматически для каждой печатной страницы.

Пример

```
<!--$_PAGEFOOTER-->  
<tr>  
<td>...</td>  
<td>...</td>  
<td>...</td>  
</tr>  
<!--$_FOOTER-->  
...
```

\$\$_FOOTER

Назначение

Начало раздела подножия всего отчета

Атрибуты

VISIBLE (exp) Выражение, возвращающее числовой результат и определяющее условие видимости всего раздела в итоговом документе. Раздел не отображается, если результат вычисления выражения равен 0. Необязательный атрибут. По умолчанию 1.

Описание

Данный тег служит для указания того, что следующий за ним текст HTML-кода является содержимым подножия отчета. Рекомендуемое расположение внутри текста страницы - до тега </TABLE> (если есть) или </BODY>.

Пример

```
<!--$$_FOOTER-->
```

\$\$_INIT

Назначение

Выполнение команд инициализации отчета

Атрибуты

Нет.

Описание

Служебный тег INIT имеет внутреннее содержание в виде последовательности строк, оканчивающихся символами <CR><LF>. Каждая строка представляет собой вызов системной или пользовательской функции, аргументам которой, в числе прочих, могут быть присвоены значения параметров, переданных отчету (см. PARAMETERS). Этот блок команд выполняется последовательно в самом начале трансляции шаблона в порядке сверху вниз. Признаком комментария внутри блока является наличие символа ";" (точка с запятой) в начале строки.

Пример

```
<!--$$_INIT  
; Следующая функция копирует значения полей одной строки таблицы во внутренний  
; именованный массив  
dMemoData("PRODUCT", "ID="+Prod_ID, "PR")  
-->
```

\$\$_INFO

Назначение

Описание настроечных параметров отчета

Атрибуты

Нет.

Описание

Служебный тег INFO имеет внутреннее содержание в виде последовательности строк, оканчивающихся символами <CR><LF>. Каждая строка представляет собой описание параметра настройки в формате: Имя=Значение. Признаком комментария внутри блока является наличие символа ";" (точка с запятой) в начале строки. Ниже приведено описание возможных имен и значений параметров настройки:

Имя	Описание
OemAnsi	Определяет тип набора символов основного источника данных. Может принимать одно из следующих значений: OEM, ANSI. По умолчанию OEM.
Permissions	Список через запятую ролей пользователей в приложении COM+/MTS, которым разрешен доступ до данного отчета. По умолчанию Все.
ZeroAsBlank	Может применяться для отмены появления в любом месте отчета нулевых числовых значений. Возможные значения: 1/0. По умолчанию 0.

Пример

```

<!--$_INFO
; Основной источник данных является DBF - таблицей FoxPro 2.5 for DOS
OEMANSI = ANSI
; Нули не отображать!
ZeroAsBlank = 1
-->

```

\$\$_PARAMETERS

Назначение

Описание параметров, используемых при получении данных для отчета

Атрибуты

Нет

Описание

Раздел описания параметров отчета относится к категории служебных и, следовательно, имеет внутреннее содержание. 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 разделе \$\$_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
-->
```

\$\$_VARS

Назначение

Раздел описания используемых для вывода элементов данных

Атрибуты

Нет

Описание

Содержание раздела VARS состоит из строк, каждая из которых описывает один элемент/выражение данных, используемое для вывода. Элемент данных описывается следующими атрибутами:

Имя	Описание
ID	Ид элемента. Значение должно совпадать с именем соответствующей закладки (bookmark) в теге типа . Если содержание атрибута ID совпадает с названием колонки набора данных, атрибут DATA может отсутствовать. Обязательный атрибут.
DATA	(exp) Выражение, результат которого выводится в итоговый отчет. В тексте выражения могут присутствовать ссылки на имена (ID) других элементов в формате @<имя_элемента>. В выражении могут применяться любые операторы и функции, описанные в разделе Выражения, функции и операторы (общие положения). Условно-обязательный атрибут (см. ID-атрибут).

FORMAT	<p>Формат вывода значения выражения. Для чисел используется следующий синтаксис: [для значения >0]~[для значения <0]~[для значения =0]~[для значения NULL]</p> <p>Необязательный атрибут.</p>
TYPE	<p>Описывает смысловое содержание элемента/выражения данных. В зависимости от значения атрибута TYPE, может понадобиться описание дополнительных атрибутов. TYPE может принимать следующие значения:</p> <ul style="list-style-type: none"> ✚ Статистическая функция (SUM,AVRG,MIN,MAX) ✚ Ссылка на ресурс (URL) ✚ Вложенный отчет (REPORT) ✚ Внутреннее свойство отчета (PROPERTY) ✚ Штрих-код (BARCODE) <p>Необязательный атрибут.</p>
HREF	(exp) Гиперссылка. Необязательный атрибут.
ALT	(exp) Альтернативная подсказка для рисунков. Необязательный атрибут.
TITLE	(exp) Всплывающая подсказка. Необязательный атрибут.
BOLD	(exp) Выделение текста. Выражение, результат которого может принимать значения 1 (выделять), 0 (не выделять). Необязательный атрибут.
OEMANSI	Вывод в формате OEM/ANSI. По умолчанию OEM. Необязательный атрибут.
XATTR	Строка текста с любыми допустимыми атрибутами элемента. Необязательный атрибут.
TARGET	Имя фрейма для вывода страницы, имя которой указано в атрибуте HREF. Необязательный атрибут.

Пример

```

<!--$$_VARS
; Вывод названия текущего предприятия выделенным шрифтом
ID=Предприятие; DATA=Dlookup$("name", "firm", "id="+pFirmID); BOLD=1
ID=ДатаСчета; DATA=doc_date; FORMAT=dd.mm.yyyy
ID=Количество; DATA=qty; FORMAT=0.#####
; В зависимости от переданного параметра выводим сумму в ценах закупки или продажи
ID=Цена; DATA=IfNull#(if(pMode=1, price_dlv, price_exp),0); FORMAT=0.00
ID=СуммаСчета; DATA=@Количество*@Цена; FORMAT=0.00
; Итоговая сумма счета
ID=ИтогоСумма; DATA=@Сумма; FORMAT=0.00; TYPE=SUM
-->
    
```

\$\$_DBCNN

Назначение

Описание источника данных

Атрибуты

Имя	Значение
ID	String. Идентификатор источника данных. Обязательный атрибут.



Описание

Внутреннее содержание дескриптора DBCONN состоит из описания параметров и их значений, необходимых для подключения к одному источнику данных (СУБД). Наличие каждого параметра не является строго обязательным и зависит от типа источника данных, указанного в параметре DBMS. Описание нового параметра должно начинаться с новой строки.

В следующей таблице приведено описание возможных параметров источника данных:

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

Если к источнику данных необходимо подключиться через ODBC-интерфейс, параметр DBParm должен содержать пару ConnectString=value.

Пример

```
<!--$$_DBS ID=001
; Подключение через ODBC-интерфейс
DBMS=ODBC
DBParm=ConnectString='DSN=TestDB; UID=guest'; SQLCache=10
-->
```

\$\$_SQL

Назначение

Описание SELECT-оператора языка SQL, или хранимой процедуры с возвращаемым набором данных.

Атрибуты

Имя	Значение
ID	String. Идентификатор SQL-запроса. Необязательный атрибут.
DBCONN	String. Идентификатор источника данных. Необязательный атрибут.

При отсутствии ссылки на внутренний источник данных макета (атрибута DBS), по умолчанию используется информация об источнике данных из параметров подключения самого OLE-сервера. Значение параметров, как и само подключение, устанавливается программно путем вызова соответствующих методов объекта.

Описание

Внутреннее содержание дескриптора 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 элементов данных в разделе VARS. Если вместо списка колонок, используется символ "*", имена колонок будут созданы автоматически на основании физической структуры таблиц(ы) базы данных.

Пример

```
<!--$$_SQL
select ID, CODE, NAME, QTY, PRICE
      from PRODUCT
      where GRP = :pGroupID
      order by << if(pMode=1, "CODE", "NAME") >> ASC
-->
```

\$\$_DONE

Назначение

Выполнение команд завершения отчета

Атрибуты

Нет.

Описание

Служебный тег DONE имеет внутреннее содержание в виде последовательности строк, оканчивающихся символами <CR><LF>. Каждая строка представляет собой вызов системной или пользовательской функции. Этот блок команд выполняется последовательно в самом конце трансляции шаблона в порядке сверху вниз. Признаком комментария внутри блока является наличие символа ";" (точка с запятой) в начале строки.

Пример

```
<!--$$_DONE
; Следующая функция очищает ресурсы, задействованные в блоке INIT
dMemoClear("PR")
-->
```

\$\$_INCLUDE

Назначение

Включение в исходный текст файла шаблона текста из другого файла

Атрибуты

Имя	Описание
FILE	(exp) Полный путь файла, встраиваемого в макет документа

Описание

В результате интерпретации данного дескриптора, текст файла, указанного в атрибуте FILE, будет "встроен" в исходный текст шаблона до начала его трансляции. Такой механизм позволяет многократно использовать в макетах документов как общие куски кода HTML (например, описание фирменного знака), так и уже написанные и отработанные программы на языках XML, JavaScript, VBScript и др.

Пример

```
<!--$$_INCLUDE FILE='lastinfo.htm'-->
```



Файл конфигурации

Поведением и инициализацией свойств сервера **REPMAKER** можно управлять, создав в той же директории, где находится `repmaker.dll`, одноименный файл с расширением `.ini`. Этот файл читается и анализируется сервером только при начальном запуске. Для того, чтобы новые изменения вступили в силу, необходим перезапуск сервера (COM+, dedicated server) или программы (in-process server).

Содержание файла `repmaker.ini` состоит из строк, каждая из которых представляет собой ключ в форме пары: `Имя=Значение`. Ключи сгруппированы по секциям, имя каждой из которых заключено в символы `[]`. В файле строки, начинающиеся символом `;` являются комментариями.

В следующей таблице описаны имена и назначение возможных секций и ключей:

Ключ	Секция	Назначение
ConnectOnActivate	Application	Integer. Дает сигнал серверу на установку соединения с сервером БД, параметры которого описаны с секции [Connection]. Может принимать следующие значения: 1/0. По умолчанию 0 (без соединения).
DstPath	Application	String. Полный путь до директории, в которой будут создаваться результирующие файлы. В описании пути могут присутствовать ссылки на переменные памяти в формате %VAR_NAME%.
SrcPath	Application	String. Полный путь до директории, в которой находятся файлы-шаблоны отчетов. В описании пути могут присутствовать ссылки на переменные памяти в формате %VAR_NAME%.
TempPath	Application	String. Полный путь до директории с временными файлами. По умолчанию этот путь совпадает со значением переменной памяти %TEMP%.
DBMS	Connection	Назначение ключа аналогично одноименному свойству объекта
DBParm	Connection	=
Database	Connection	=
ServerName	Connection	=
LogID	Connection	=
LogPass	Connection	=
UserID	Connection	=

DBPass	Connection	=
Lock	Connection	=

COM – интерфейс

Сервисы, предоставляемые генератором HTML-отчетов RepMaker, 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.HTML**) 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 Приложении 2 данного руководства, а также в директории \SAMPLES, созданной после инсталляции REPMAKER.

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

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)

CheckNavigateBefore()	Функция контроля параметра URL, которая может применяться в событии BeforeNavigate2 (Navigating) элемента управления Microsoft WebBrowser Control
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
Evaluate()	Вычисление некоторого выражения в контексте заданной строки
GetResultSet()	Получение объекта типа ResultSet (ADODB RecordSet)
IsConnected()	Проверка на наличие соединения с базой данных
Load()	Загрузка файла-шаблона с разбором (parsing) его составляющих
MakeFile()	Создание результирующего файла, на основе шаблона, загруженного с помощью функции Load()
MakeHTML()	Генерация текстовой строки, содержащей HTML-скрипт выходного файла
SaveAs()	Сохранить файл отчета в заданном формате
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
Translate()	Создание текста выходного документа на основе заданного имени шаблона

Передача параметров отчету

Имена, типы и значения по умолчанию параметров отчета описываются в специальном разделе шаблона 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.MakeFile("c://temp//res0001.htm?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.htm ? #1=" + 120 + "&#2='10.01.2003' & #3=NULL" )
```

Далее идет более подробное описание методов COM-объекта:

Метод

```
CheckNavigateBefore(url: String, flags: Integer, frame: String,  
readystate: Integer, REF cancel: Boolean): String
```

Назначение

Контроль адреса WEB-ресурса при щелчке на какой-либо гиперссылке текущей HTML-страницы.

Аргументы

Url	String [in]. Адрес WEB-ресурса, определенный для гиперссылки
Flags	Integer [in]. Битовая маска служебных значений элемента WebBrowser.
Frame	String [in]. Название фрейма, в котором должна быть отображена новая страница
ReadyState	Integer[in]. Числовое значение, выражающее состояние объекта WebBrowser.
Cancel	Boolean [out]. Логическое значение, возвращаемое методом и означающее необходимость блокирования процесса вывода страницы (True) или его продолжение (False). В таких языках программирования, как PowerBuilder и C#, перед именем аргумента должно присутствовать слово REF.

Возвращаемое значение

String. Модифицированный адрес URL для страниц со специальным протоколом `her://`. Иначе пустая строка (`""`).

Описание

Данный метод объекта может быть использован в событии `NavigateBefore2` (`Navigating`) визуального элемента управления *Microsoft WebBrowser Control*, предназначенного для отображения HTML-страниц в клиентском приложении. Метод возвращает непустое значение только для страниц с протоколом `her://`, а также присваивает значение `True` аргументу `Cancel`, переданному по ссылке.

Примеры

```
Private Sub WBC1_BeforeNavigate2(ByVal pDisp As Object, URL As Variant,  
flags As Variant, TargetFrameName As Variant,  
PostData As Variant, Headers As Variant, Cancel As Boolean)  
  
Dim f As String  
f = oRMK.checknavigatebefore(URL, flags, TargetFrameName, WBC1.ReadyState, Cancel)  
If f <> "" Then  
    ` Делаем переход по "правильному" адресу  
    WBC1.Navigate f, flags, TargetFrameName, PostData, Headers  
End If  
End Sub
```

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()
...
```

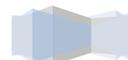
```
myfile.ini
-----
[Connection]
DBMS=ODBC
DBParm=ConnectionString='DSN=MyDSN; UID=MyUserName'
; RU (Read Uncommitted)
Lock=RU
```

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

Evaluate(script:String, row:Long): String

Назначение

Вычисление результата выражения в контексте указанного номера строки из набора данных

Аргументы

script	String [in]. Строка, содержащая некоторое выражение, результат которого возвращается методом
row	Long [in]. Номер записи выборки данных, над которой вычисляется выражение, указанное в аргументе <i>script</i>

Возвращаемое значение

String. Результат вычисления выражения в виде символьного значения. Если выполнение выражения *script* привело к ошибке по различным причинам (синтаксическая ошибка, неверно указаны имена колонок и пр.), возвращается символ "!".

Описание

Метод вычисляет значение выражения, заданного в аргументе *script*, над строкой данных, номер которой указан в аргументе *row*. В выражении могут быть задействованы все операторы и функции, описанные в разделах Базовые функции и Функции расширения. Кроме того, в выражении могут участвовать названия колонок и вычисляемых значений текущего набора данных.

Примеры

```
// PowerBuilder  
double nRes  
nRes = Double( oRepMaker.Evaluate("sum(price*qty for all)", 1) )
```

Метод

GetResultSet(ref rs:ResultSet): Integer

Назначение

Получение объекта типа ResultSet (ADODB RecordSet)

Аргументы

rs	ResultSet [out]. Объект, тип которого совместим с типом ADODB RecordSet, и переданный по
----	--

ссылке.

Возвращаемое значение

Integer. Метод возвращает 1 в случае успешного присвоения, и -1 в противном случае.

Описание

Метод предназначен для создания и присвоения ссылки на объект типа ADODB RecordSet. В случае успешного выполнения, объект *resultset* будет ссылаться на внутреннюю таблицу с данными COM-объекта, созданную на основании выполненного SQL-запроса. При этом обработка данных может осуществляться и с применением полученного объекта *resultset*.

Примеры

```
' Visual Basic
dim oRM as RepMaker.Application
dim oRS as ADODB.RecordSet
dim rc as Integer
rc = oRM.Generate("c:\templ\templ_01.htm")
if oRM.GetResultSet(oRS) = 1 then
    rst.MoveFirst
    With oRs
    Do While Not .EOF
        Debug.Print .Fields(0), .Fields(1), .Fields(2), .Fields(4)
        .MoveNext
    Loop
    End With
end if
```

Метод

Load(filename:String) : Integer

Назначение

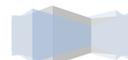
Загрузка файла-шаблона с разбором (parsing) его составляющих

Аргументы

filename	String [in]. Путь до файла-шаблона отчета. 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 методами MakeHTML() или MakeFile().

Примеры

```
// PowerBuilder
string ls_html
if oRM.Load("c:\templ\t001.htm") = 1 then
    oRM.SetParams({"01.07.2003", "10.07.2003"})
    ls_html = oRM.MakeHTML()
    oWebBrowser.object.Document.Write( ls_html )
end if
```

Метод

MakeFile(filename:String): Integer

Назначение

Создание результирующего файла, на основе шаблона, загруженного с помощью функции Load()

Аргументы

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

Возвращаемое значение

Integer. При успешном создании файла отчета, метод возвращает 1. Иначе -1.

Описание

Данный метод генерирует результирующий файл отчета на основании предварительно выполненного метода Load() COM-объекта.

Примеры

```
// JavaScript
<script defer>
function GenReport(f, p1, p2) {
if oRM.Load("c:\templ\t0001.htm") = 1
{
    oRM.MakeFile(f+"?#1="+p1+"&#2="+p2);
    window.open( f );
}
}
</script>
```

Метод

MakeHTML(): String

Назначение

Генерация текстовой строки, содержащей HTML-скрипт выходного файла

Аргументы

Нет

Возвращаемое значение

String. Текст HTML сгенерированного выходного файла

Описание

Метод генерирует текст выходного файла отчета на основании результата, полученного при последнем выполнении метода Load() COM-объекта.

Примеры

```
' Visual Basic
dim sHTML as string
if oRM.Load("c:\templ\t001.htm") = 1 then
    oRM.SetParam("pEmpID", 100)
    sHTML = oRM.MakeHTML()
    oWBC.document.write(sHTML)
end if
```

Метод

SaveAs(htmlfile: String, astype: String): String

Назначение

Сохранение html-файла в заданном формате

Аргументы

Htmlfile String. Полный путь до html-файла

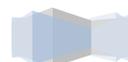
Astype String. Вид создаваемого файла. Может принимать следующие значения: Word!, Excel!

Возвращаемое значение

String. Полный путь до созданного файла. В случае ошибки возвращается "". Информацию об ошибке можно узнать через свойства объекта *ErrCode* и *ErrText*.

Описание

Метод осуществляет сохранение указанного html-файла в формате, заданном в аргументе *AsType*.



Примеры

Visual Basic

```
Dim hf as String
Dim wf as String

oRMK = CreateObject("REPMAKER.HTML")
if oRMK.Load(АнализПрихода.htm) = 1 then
    hf = oRMK.MakeFile("АнализПрихода.htm" )
    if hf <> "" then
        wf = oRMK.SaveAs(hf, "Word!")
    end if
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

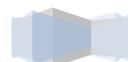
```
oRMK = CreateObject("REPMAKER.HTML")
oRMK.SetDstPath("%TEMP%\HTMLREP")
```

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.htm"
if oRMK.Load("d:\templates\"+ls_filename) = 1 then
    oRMK.SetParam("pFirmID", 1)
    if oRMK.MakeFile("c:\temp\"+ls_filename) = 1 then
        inv_inet.HyperLinkToURL("c:\temp\"+ls_filename)
    else
        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.htm"
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.MakeFile("c:\temp\"+ls_filename) = 1 then
        inv_inet.HyperLinkToURL("c:\temp\"+ls_filename)
    else
        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.HTML")
oRMK.SetSrcPath("C:\REPSRV\TEMPLATES")
```

Метод

Translate(filename:String, type:String): String



Назначение

Создание текста выходного документа на основе заданного имени шаблона

Аргументы

filename String [in]. Путь до файла шаблона
type String [in]. Тип преобразования: HtmlFile!, HtmlText!.

Возвращаемое значение

String. Смысловое значение возвращаемой строки символов зависит от значения аргумента *type*. В случае ошибки возвращается пустая строка.

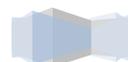
Значение <i>type</i>	Возвращаемое значение содержит
HtmlFile!	Полный путь до результирующего файла отчета. Информация о пути берется на основе значения свойства <i>DstPath</i>
HtmlText!	Текст отчета в формате HTML

Описание

Метод выполняет преобразование файла макета в ресурс определенного параметром *type* вида.

Примеры

```
WebBrowser.Navigate (oRepMaker.Translate("start.htm", "HtmlFile!"))
```



Приложение 1. Образец шаблона

```

<html>
<head>
<meta NAME="author" content="Стрелков А.В., ООО 'ЮНИСОФТ'">
<meta http-equiv="Content-Language" content="ru">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Карточка аналитического счета</title>
<!--$$_PARAMETERS
ID=DateBeg;   DATATYPE=datetime;  DEFAULT=@V_DATE1;
ID=DateEnd;   DATATYPE=datetime;  DEFAULT=@V_DATE2;
ID=ContoCode; DATATYPE=string;    DEFAULT=@V_CODE;
ID=AContoId;  DATATYPE=number;    DEFAULT=@V_ACONTO
ID=BOW;       DATATYPE=number;    DEFAULT=@BOW
ID=FirmID;    DATATYPE=number;    DEFAULT=@FIRM_ID
-->
<!--$$_SQL
SELECT 0 as side, conto_b.code as corr_code, SUM( "mainbook"."summa") as summa
FROM mainbook, conto conto_a, conto conto_b
WHERE ( "mainbook"."db_conto" = "conto_a"."id" ) and
      ( "mainbook"."cr_conto" = "conto_b"."id" ) and
      ( ( "mainbook"."accdate" between :DateBeg and :DateEnd ) AND
        (conto_a.code = :ContoCode OR conto_a.code LIKE :ContoCode|'%.%' ) AND
        ("mainbook"."db_aconto" = :AContoID) AND
        ( "mainbook"."BOW" in (:bow,3)) AND
        ("mainbook"."FIRM" = :FirmID) )
GROUP BY corr_code
UNION
SELECT 1 as side, conto_a.code as corr_code, SUM( "mainbook"."summa")
FROM mainbook, conto conto_a, conto conto_b
WHERE ( "mainbook"."db_conto" = "conto_a"."id" ) and
      ( "mainbook"."cr_conto" = "conto_b"."id" ) and
      ( ( "mainbook"."accdate" between :DateBeg and :DateEnd ) AND
        (conto_b.code = :ContoCode OR conto_b.code LIKE :ContoCode|'%.%' ) AND
        ("mainbook"."cr_aconto" = :AContoID) AND
        ( "mainbook"."BOW" in (:bow,3)) AND
        ("mainbook"."FIRM" = :FirmID) )
GROUP BY corr_code ORDER BY 1, 2
-->
<!--$$_INIT
; Блок команд инициализации отчета
VarSet$("НачСальдо", Acc$Saldo(ContoCode+"#" + AContoId, DateBeg, FirmID))
-->
<!--$$_VARS
; Блок описания переменных для подстановки результата выражения вместо соответствующего тега <A
name="ID"></A>
ID=ВхСальдо; DATA=VarGet$("НачСальдо")
ID=ИсхСальдо; DATA=Acc$OutSaldo(ContoCode+"#" + AContoId, @СальдоВхДБ, @СальдоВхКР, @ОборотДБ, @ОборотКР)
ID=Предприятие; DATA=DLookUp$("name", "firm", "id" + FirmID);
ID=Счет; DATA=ContoCode + ' (' + Acc$ContoName(ContoCode) + ') '
ID=АналитикаСчета; DATA=Acc$ContoName("#" + AContoID)
ID=ДатаНачалаПериода; DATA=DateBeg; FORMAT=dd.mm.yyyy
ID=ДатаОкончанияПериода; DATA=DateEnd; FORMAT=dd.mm.yyyy
ID=СальдоВхДБ; DATA=StrPart#(@ВхСальдо, 1, "~t"); FORMAT=#, ##0.00
ID=СальдоВхКР; DATA=StrPart#(@ВхСальдо, 2, "~t"); FORMAT=#, ##0.00
ID=Сторона; DATA=if(side=0, 'По дебету счета '+ContoCode+' с кредита счетов', 'По кредиту счета
'+ContoCode+' с дебета счетов')
ID=НазваниеКорпСч; DATA=Acc$ContoName(corr_code)
ID=КорпСч; DATA=corr_code
ID=СуммаКорреспонденции; DATA=summa; FORMAT=#, ##0.00~#, ##0.00~#
ID=ИтогоСумма; DATA=summa; FORMAT=#, ##0.00~#, ##0.00~#; TYPE=SUM;
ID=ОборотДБ; DATA=SUM(if(side=0, summa, 0)); FORMAT=#, ##0.00~#, ##0.00~#; TYPE=SUM;
ID=ОборотКР; DATA=SUM(if(side=1, summa, 0)); FORMAT=#, ##0.00~#, ##0.00~#; TYPE=SUM;
ID=СальдоИсхДБ; DATA=StrPart#(@ИсхСальдо, 1, "~t"); FORMAT=#, ##0.00
ID=СальдоИсхКР; DATA=StrPart#(@ИсхСальдо, 2, "~t"); FORMAT=#, ##0.00
-->
</head>
<body>
<!--$$_HEADER-->
<p align="left">Предприятие: <a name="Предприятие"></a></p>
<h3 align="center" style="margin-top: 0; margin-bottom: 0">Карточка аналитического счета</h3>

```

```

<h4 align="left" style="margin-top: 0; margin-bottom: 0">Счет: <a name="Счет"></a> </h4>
<h4 align="left" style="margin-top: 0; margin-bottom: 0">Аналитика:
<a name="АналитикаСчета"></a></h4>
<h4 align="left" style="margin-top: 0; margin-bottom: 0">Период с
<a name="ДатаНачалаПериода"></a> по <a name="ДатаОкончанияПериода"></a></h4>
<p align="center" style="margin-top: 0; margin-bottom: 0"> </p>
<table width="48%" border="1" cellpadding="0" cellspacing="0">
<tr>
<td rowspan="2" width="45%"> <p align="center">Сальдо входящее </p> </td>
<td width="26%"> <p align="center">Дебет</p> </td> <td width="29%"><p align="center">Кредит</p></td>
</tr>
<tr>
<td width="26%"> <p align="center"><a name="СальдоВхДБ"></a></p></td>
<td width="29%"> <p align="center"><a name="СальдоВхКР"></a></p></td>
</tr>
</table>
<p style="margin-top: 0; margin-bottom: 0"> </p>
<table border="1" width="100%" cellspacing="0" cellpadding="0">
<tr>
<td colspan="2" bgcolor="#CCCCCC" width="594"><p align="center"><b> Корреспондирующий счет </b></td>
<td width="156" rowspan="2" bgcolor="#CCCCCC"><p align="center"><b> Сумма</b> <p align="center"
STYLE="margin-top: 0"><b> корреспонденции </b></td>
</tr>
<tr>
<td width="457" align="center" bgcolor="#CCCCCC"><b> Название </b></td>
<td width="135" align="center" bgcolor="#CCCCCC"><b> Счет (субсчет) </b></td>
</tr>
<!--$$ _GROUPHEAD#1 GROUPBY=side-->
<tr>
<td colspan="3" width="752" bgcolor="#FFFF00"><b> <a name="Сторона"></a></b></td>
</tr>
<!--$$ _DETAIL-->
<tr>
<td width="457"> <a name="НазваниеКорпСч"></a> </td>
<td width="135"> <p align="center"><a name="КорпСч"></a></p> </td>
<td width="156"> <p align="right"><a name="СуммаКорреспонденции"></a></p> </td>
</tr>
<!--$$ _GROUPFOOT#1-->
<tr>
<td colspan="2" width="594"> <p align="right"><b> Итого : </b> </td>
<td width="156"> <p align="right"><b><a name="ИтогоСумма"></a></b></p> </td>
</tr>
<!--$$ _FOOTER-->
</table>
<p style="margin-top: 0; margin-bottom: 0"> </p>
<table width="48%" border="1" cellpadding="0" cellspacing="0">
<tr>
<td rowspan="2" width="43%"> <p align="center">Сальдо исходящее</p> </td>
<td width="28%"> <p align="center">Дебет</p> </td>
<td width="29%"> <p align="center">Кредит</p> </td>
</tr>
<tr>
<td width="28%"> <p align="center"><a name="СальдоИсхДБ"></a></p>
</td> <td width="29%"> <p align="center"><a name="СальдоИсхКР"></a></p> </td>
</tr>
</table>
<p style="margin-top: 0; margin-bottom: 0"> </p>
<p style="margin-top: 0; margin-bottom: 0"> </p>
<p style="margin-top: 0; margin-bottom: 0">Главный бухгалтер
</p>
</body>
</html>

```



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.HTML")
    if rc < 0 then
        MessageBox("Connection Error", "Error number "+string(rc))
        Return
    end if
    // Параметры соединения с БД (database connection string)
    oRML.DBMS = "ODBC"
    oRMK.DBParm = "ConnectionString='DSN=TestDB;UID=Empl'"
    oRMK.DbConnect()
    // Загрузка HTML-шаблона (loading HTML-template)
    rc = oRMK.Load("c:\templates\templ1.htm")
    // Установка значения параметра (set parameter's value)
    rc = oRMK.SetParam("FirmID", 5)
    // Генерация отчета (Report generation)
    oRMK.makeFile("c:\tmp\out.htm")
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.HTML")
If Not IsObject(oRMK) Then Return
res = oRMK.Load("c:\templates\templ01.htm")
res = oRMK.SetParams(pars)
oRMK.DBMS = "ODBC"
oRMK.DBParm = "ConnectionString='DSN=TestDB'"
oRMK.DbConnect()
oRMK.MakeFile "c:\tmp\out.htm"
Shell ("explorer.exe c:\tmp\out.htm")
End Sub
```

JavaScript

```
<html>
<head>
<!------- RepMaker COM-object ----->
```



```
<OBJECT ID="RepMaker" viewastext STYLE="display:none"
        CLASSID="CLSID:AE98E08E-3F5D-43AC-ACA3-2A6118A27B50"
</OBJECT>
<!------->
<script difer>
var s;
s = RepMaker.Translate("c:\\templates\\templ2.htm", "html");
document.open("text/html", "replace");
document.write(s);
document.close();
</script>

</head>
<body>

</body>
</html>
```

