

## Приложение Е – файл EXPORT.INI

При экспорте данных из программы RGS в текстовые файлы, для описания форматов экспорта, используется файл EXPORT.INI. Структура файла аналогична стандартной структуре INI-файлов для WINDOWS. Одному формату соответствует одна группа, состоящая из восьми строк, далее идет пустая строка и за ней группа с описанием следующего формата и т.д.

Правила для составления группы с описанием формата:

### **[Name\_Format]**

В первой строке группы, в квадратных скобках, описывается название формата которое заносится в список форматов в диалоговом окне **Формат экспорта**.

Например - **[Каталог пунктов RGD v5.0]** – в диалоговом окне **Формат экспорта**, в списке будет строка «Каталог пунктов RGD v5.0».

### **Map=**

Через запятую описываются названия заголовков таблицы диалогового окна **Формат экспорта**.

Например - **Map= Пункт, X, Y, H, Код, № п/п** – в диалоговом окне **Формат экспорта**, в заголовках таблиц будут перечисленные значения.

### **Header=**

В кавычках описывается заголовок файла (текст в начале файла). Допускается использовать специальные символы: \n – для определения новой строки; \t – для вставки табуляции.

Например - **Header="RGD V5.0\n\n<CP"** – в начале экспортируемого файла будет следующая запись:

```
RGD v5.0
```

```
<CP
```

### **Footer=**

В кавычках описывается окончание файла.

Например - **Footer="<LG"** – в конце экспортируемого файла, после записи данных, будет следующая запись:

```
<LG
```

### **Extension=**

Описывается расширение файла экспорта присваиваемое ему по умолчанию.

Например - **Extension=rgd** – в стандартном диалоговом окне открытия файла экспорта, в качестве расширения файла, по умолчанию будет предложено: \*.rgd

### **Format=**

Описывается формат строки данных. Формат строки аналогичен формату функции PRINTF языка программирования Си (Описание функции PRINTF приводится ниже). Количество и порядок данных указывается в строке «Order=»

Например - **Format="%-12s %14.31f %14.31f %10.31f %s"**

### **Order=**

Описывается количество и порядок колонок в таблице диалогового окна **Формат экспорта** и соответственно в экспортируемом файле. Каждому элементу данных соответствует номер

1 – название пункта или точки

2 – координата X

3 – координата Y

4 – отметка H

5 – код

6 – порядковый номер

Например - **Order=1,2,3,4,5** – в совокупности с данными в строке «Format=», данные в экспортируемом файле будут выглядеть следующим образом:

кол1	630.198	355.406	0.000	1195
кол2	613.663	380.787	152.622	1195
кол3	602.632	374.068	154.531	1195

### **Sort=**

Указывается номер данных, по которым производится сортировка в экспортируемом файле. Если номер не указан, данные сортируются в порядке ввода, если перед номером стоит знак минус '-', данные сортируются в обратном порядке.

## Описание функции PRINTF

Функция *printf* печатает символы, формирует и печатает задаваемые аргументами значения в стандартный выводной поток *stdout* (ввод/вывод верхнего уровня). Функция *printf* имеет переменное число параметров.

Строка описания формата вывода, адрес которой задается значением единственного обязательного параметра *format\_string*, состоит из обычных символов, специальных управляющих последовательностей символов (*escape*-последовательностей) и, если за параметром *format\_string* следуют еще дополнительные аргументы, спецификаций полей формата вывода по одному для каждого дополнительного аргумента.

Обычные символы и *escape*-последовательности просто копируются в поток *stdout* в порядке их появления. Например, оператор

```
printf(" Первая строка\n\t\t Вторая строка\n");
```

выводит:

Первая строка

Вторая строка

Если за параметром *format\_string* следуют аргументы *arguments*, то *format\_string* должна содержать спецификации форматов, определяющих выводной формат для этих аргументов.

Спецификации форматов начинаются с символа процента ('%') и описываются ниже.

Строка описания формата (адрес которой задается значением параметра *format\_string*) прочитывается слева направо. Когда встречается первая спецификация формата, значение первого аргумента после параметра *format\_string* преобразуется и выводится согласно спецификации формата. Вторая спецификация формата преобразует и выводит второй аргумент, и, таким образом, обработка продолжается до конца *format\_string*. Если задано больше аргументов, чем спецификаций формата, лишние аргументы игнорируются.

Результат будет неопределенным, если нет достаточного количества аргументов для всех спецификаций форматов. Спецификация формата имеет следующую форму:

```
% [flags] [width] [.precision] [F|N|h|l|L] type
```

Спецификация формата не содержит внутри себя пробелов. Каждое поле спецификации формата есть одиночный символ или число, обозначающие обязательный параметр формата. Символ *type*, который появляется после последнего поля формата, определяет, как будет интерпретироваться соответствующий данному описанию формата аргумент: как символ, строка или число (смотри таблицу 1).

Самая простая спецификация формата содержит только знак и символ *type* (например, "%s").

Необязательные поля управляют другими параметрами форматирования:

<i>flags</i>	выравнивание выводных символов, управление печатью знаковых символов ('+' и '-'), пробелов, десятичных точек, восьмеричных и шестнадцатеричных префиксов (смотри таблицу 2)
<i>width</i>	минимальное число выводимых символов
<i>precision</i>	максимальное число символов, которые будут напечатаны, для всех или части выводных полей; или минимальное число цифр, которые будут печататься, для значения целого (смотри таблицу 3).
<i>F, N</i>	префиксы, которые позволяют пользователю не принимать во внимание способы адресации и используемую модель памяти: <i>F</i> используется в малой модели для печати значений, которые описаны как <i>far</i> <i>N</i> используется в средней, большой и верхней моделях для значений <i>near</i> префиксы <i>F</i> и <i>N</i> должны быть использованы только с типами "%s" и "%p", так как они имеют смысл только с аргументами, задаваемыми как указатель.
<i>h, l</i>	размер аргумента: <i>h</i> используется как префикс с целыми типами <i>d, i, o, u, x</i> и <i>X</i> , чтобы определить, что аргумент является коротким целым ( <i>short int</i> ). <i>l</i> используется как префикс с типами <i>d, i, o, u, x</i> и <i>X</i> , чтобы определить, что аргумент является длинным целым ( <i>long int</i> ); также используется как префикс с типами <i>c, e, f, g</i> или <i>G</i> , чтобы показать, что аргументы имеют тип <i>double</i> , а не <i>float</i> .

Если за символом процента ('%') следует символ, который не является полем формата, этот символ просто копируется в *stdout*.

Таблица 1.

Символы <i>type</i>		
Символ	Тип аргумента	Выводной формат
d	целое	десятичное целое со знаком
i	целое	десятичное целое со знаком
u	целое	десятичное целое без знака
o	целое	восьмеричное целое без знака
x	целое	шестнадцатеричное целое без знака, для обозначения шестнадцатеричных цифр используются символы "abcdef"
X	целое	шестнадцатеричное целое без знака, для обозначения шестнадцатеричных цифр используются символы "ABCDEF"
f	с плавающей точкой	Значение со знаком, имеет форму [-] dddd.dddd, где dddd - одна и более десятичных цифр. Число цифр до десятичной точки зависит от величины значения, а число цифр после десятичной точки зависит от требуемой точности.
e	с плавающей точкой	Значение со знаком имеет форму [-] d.dddde [sign] ddd, где d - одна десятичная цифра, dddd - одна или более десятичных цифр, ddd - ровно три десятичные цифры, и sign - это '+' или '-'.
E	с плавающей точкой	Идентично 'e' формату, но знак 'e' заменяется на 'E'
g	с плавающей точкой	Значение со знаком печатается в формате 'f' или 'e', однако является более компактным для заданного значения и точности <i>precision</i> (смотри ниже). Формат 'e' используется только тогда, когда экспонента значения меньше, чем -4, или больше, чем <i>precision</i> . Незначимые нули отбрасываются, десятичная точка печатается только в том случае, если одна или более цифр следуют за ней.
G	с плавающей точкой	Идентично формату 'g', но символ 'E' используется для обозначения экспоненты вместо 'e'
c	символ	Одиночный символ.
s	строка	Символы печатаются до первого нулевого символа ('\0') или до тех пор, пока не будет напечатано <i>precision</i> символов.

Таблица 2

Символы <i>flag</i>		
Флаг	Действие флага	По умолчанию
-	Результат, если количество символов для вывода меньше указанного размера поля ( <i>width</i> ), выравнивается по левой границе (дописываются пробелы справа)	Выравнивание по правой границе
+	Добавляет префикс знака ('+' или '-'), для выводного значения, если выводное значение является знаковым типом;	Знак печатается только для отрицательных значений.
Пробел	Добавляет префикс пробел для выводного значения, если выводное значение со знаком и положительно; '+' преобладает над флагом ' ', если указаны оба.	Пробел не добавляется
#	Когда используется формат <i>o</i> , <i>x</i> или <i>X</i> , флаг '#' добавляет префикс ненулевому выводному значению: <i>0</i> , <i>0x</i> или <i>0X</i> соответственно. Когда используется формат <i>e</i> , <i>E</i> или <i>f</i> , выводное значение будет содержать десятичную точку во всех случаях. Когда используется формат <i>g</i> или <i>G</i> , выводное значение будет содержать десятичную точку и незначимые нули не подавляют	Нет префикса.

Поле *width* является неотрицательным десятичным целым, управляющим минимальным числом символов, выводимых по данному описателю формата. Если число символов в выводном значении меньше, чем определено в *width*, пробелы добавляются слева или справа (в зависимости от того, определен ли флаг '-') до тех пор, пока не будет достигнут размер *width*.

Спецификация *width* никогда не означает обрезание значащих символов в выводимом значении; если число символов в выводном значении больше, чем определено в *width*, или *width* не задано, печатаются все символы (возможно, согласно указаниям спецификации *precision*).

Спецификация *width* может быть задана как символ звездочка ('\*'); в этом случае сам текущий аргумент из списка аргументов (который должен иметь тип *int*) предполагается как значение, задающее минимальное количество выводимых символов. Этот аргумент должен предшествовать в списке аргументов значению, которое будет форматироваться и выводиться по текущему описателю формата.

Спецификация *precision* является неотрицательным десятичным числом, следующим за точкой ( '.' ), она определяет точное число символов, которые должны быть напечатаны, или место десятичной точки.

Спецификация *precision* может сузить выводное значение или округлить в случае значения с плавающей точкой.

Спецификация *precision* может быть задана символом звездочка ('\*'); в этом случае сам аргумент из списка аргументов предполагается как значение *precision*. Аргумент *precision* должен предшествовать в списке аргументов значению, которое будет форматироваться.

Интерпретация значения *precision* в зависимости от заданного типа *type* показана в таблице 3.

Таблица 3

Тип	Действие	По умолчанию
d i u o x X	<i>Precision</i> определяет минимальное число цифр, которые будут напечатаны. Если число цифр в аргументе меньше, чем <i>precision</i> , выводное значение добавляется слева нулями. Значение не сжимается, когда число цифр превышает <i>precision</i> .	Если <i>precision</i> есть 0, или отсутствует, или точка появилась без числа, следующего за ним, <i>precision</i> устанавливается в 1.
e E f	<i>Precision</i> определяет число цифр, которые будут напечатаны после десятичной точки. Последняя напечатанная цифра округляется.	По умолчанию <i>precision</i> равно 6. Если <i>precision</i> равно 0, десятичная точка не печатается.
g G	<i>Precision</i> определяет максимальное число значимых цифр, которые будут напечатаны.	Все десятичные цифры печатаются.
c	Нет действия	Символ печатается.
s	<i>Precision</i> определяет максимальное число символов, которые будут напечатаны. Лишние символы не печатаются.	Символы печатаются до тех пор, пока не будет достигнут символ конца строки ('\\0')