搜索



# 替换函数

QMAKE提供了在配置过程中处理变量内容的函数。这些函数称为*替换函数*。通常，它们返回可分配给其他变量的值。您可以通过在函数前面加上运算符来获取这些值。替换函数可以分为内置函数和函数库。$$

另请参阅测试函数。

## 内置替换功能

基本替换函数作为内置函数实现。

### absolute_path（路径[，基]）

返回 的绝对路径。path

未指定 ifis，使用当前目录作为基目录。如果是相对路径，则在使用前相对于当前目录解析。base

例如，以下调用返回字符串："/home/johndoe/myproject/readme.txt"

```
message($$absolute_path("readme.txt", "/home/johndoe/myproject"))
```

此功能在Qt 5.0中引入。

另请参阅clean_path（），relative_path（）。

### 基本名称（变量名称）

返回 中指定的文件的基名。variablename

例如：

```
FILE = /etc/passwd
FILENAME = $$basename(FILE) #passwd
```

- blob将文件的全部内容作为一个值返回
- lines将每一行作为单独的值返回（不带行尾）
- true（默认值）并将文件内容作为单独的值返回，根据 qmake 值列表拆分规则进行拆分（如在变量赋值中）。lfis，仅包含换行符的值将插入到列表中，以指示换行符在文件中的位置。falsemodefalse

## clean_path（路径）

返回时，目录分隔符规范化（转换为"/"）并删除冗余分隔符，以及""s 和 ".."已解决（尽可能）。此函数是 QDir：：cleanPath 的包装器。path

此功能在Qt 5.0中引入。

另见absolute_path（），relative_path（），shell_path（），system_path（））。

## 目录（文件）

返回指定文件的目录名称部分。例如：

```
FILE = /etc/X11R6/XF86Config
DIRNAME = $$dirname(FILE) #/etc/X11R6
```

## enumerate_vars

返回所有已定义的变量名称的列表。

此功能在Qt 5.0中引入。

## escape_expand（arg1 [， arg2 ...， argn]）

接受任意数量的参数。它为每个参数扩展转义序列，并将参数作为列表返回。\n\r\t

> **注意：**如果指定要按字面方式扩展的字符串，则需要转义反斜杠，如以下代码片段所示：

```
message("First line$$escape_expand(\\n)Second line")
```

## find（variablename， substr）

返回与正则表达式匹配的所有值。variablenamesubstr

```
MY_VAR = one two three four
MY_VAR2 = $$join(MY_VAR, " -L", -L) -Lfive
MY_VAR3 = $$member(MY_VAR, 2) $$find(MY_VAR, t.*)
```

## files（pattern[， recursive=false]）

展开指定的通配符模式并返回文件名列表。如果是真的，这个函数下降到子目录。recursive

## 第一个（变量名）

返回 的第一个值。variablename

例如，以下调用返回：firstname

```
CONTACT = firstname middlename surname phone
message($$first(CONTACT))
```

另请参阅take_first（），last（）。

## format_number（数字[，选项...]）

以指定的格式返回。您可以指定以下选项：numberoptions

- ibase=n将输入的基数设置为n
- obase=n将输出的基数设置为n
- width=n将输出的最小宽度设置为如果输出短于，则用空格填充nwidth
- zeropad用零而不是空格填充输出
- padsign在输出中为正值预置空格
- alwayssign在输出中为正值加上加号
- leftalign将填充放在输出中值的右侧

目前不支持浮点数。

例如，以下调用将十六进制数转换为：BAD002989

```
message($$format_number(BAD, ibase=16 width=6 zeropad))
```

此功能在Qt 5.0中引入。

## fromfile（文件名、变量名）

计算为 qmake 项目文件并返回分配给的值。filenamevariablename

另请参阅infile（）。

## getenv（变量名）

返回环境变量的值。这主要等同于语法。但是，该函数支持名称中带有括号的环境变量。
variablename$$(variablename)getenv

连接（变量名、粘附、之前、之后）

联接的值。如果此值不为空，则此函数在值前面加上前缀，并在唯一的必填字段 with.is 后缀，其他字段默认为空字符串。如果需要在 中编码空格，或者，则必须用引号括起来。
`variablenamegluebeforeaftervariablenamegluebeforeafter`

## 最后（变量名）

返回 的最后一个值。`variablename`

例如，以下调用返回：phone

```
CONTACT = firstname middlename surname phone
message($$last(CONTACT))
```

另请参阅take_last（），first（）。

## list（arg1 [， arg2 ...， argn]）

采用任意数量的参数。它创建一个唯一命名的变量，其中包含参数列表，并返回该变量的名称。您可以使用该变量编写循环，如以下代码片段所示

```
for(var, $$list(foo bar baz)) {
    ...
}
```

而不是：

```
values = foo bar baz
for(var, values) {
    ...
}
```

## lower（arg1 [， arg2 ...， argn]）

采用任意数量的参数并将其转换为小写。

另请参阅上（）。

## 成员（变量名 [， 开始 [， 结束]]）

返回列表值的切片，其中从零开始的元素索引介于 and（（含）之间。`variablenamestartend`

ifis 未给出，它默认为零。此用法等效于。`start$$first(variablename)`

ifis 不给出，它默认为。这种用法表示简单的数组索引，因为将只返回一个元素。`endstart`

也可以在单个参数中指定开始和结束，数字中两个句点分隔

如果任一索引超出范围，则返回空列表。

如果小于，则元素以相反的顺序返回。endstart

> **注意：**结束索引是包含性和无序的事实意味着仅当索引无效时才返回空列表（输入变量为空所暗示）。

另请参阅str_member（）。

## num_add（arg1 [， arg2 ...， argn]）

获取任意数量的数值参数并将它们相加，返回总和。

隐式支持减法，因为可以简单地在数值前面加上减号来取反：

```
sum = $$num_add($$first, -$$second)
```

如果操作数可能已经为负数，则需要执行另一个步骤来规范化数字：

```
second_neg = -$$second
second_neg ~= s/^--//
sum = $$num_add($$first, $$second_neg)
```

此功能在Qt 5.8中引入。

## 提示（问题 [，装饰]）

显示指定的值，并返回从标准输入读取的值。question

If is*true*（默认值），问题获得一个通用前缀和后缀，将其标识为提示。decorate

## 报价（字符串）

将整体转换为单个实体并返回结果。这只是将字符串括成双引号的一种奇特方式。string

## re_escape（字符串）

返回每个使用反斜杠转义的特殊正则表达式字符。此函数是QRegularExpression：：escape 的包装器。string

## read_registry（树、键[、标志]）

返回树内注册表项的值。keytree

仅支持 trees（） 和 （）。HKEY_CURRENT_USERHKCUHKEY_LOCAL_MACHINEHKLM

可能是（） 或（）。flagWOW64_32KEY32WOW64_64KEY64

此功能在Qt 5.12.1中引入。

## relative_path（filePath[， base]）

返回相对于 的路径。filePathbase

未指定 ifis，它是当前项目目录。如果是相对的，则在使用前相对于当前项目目录进行解析。base

ifis 相对的，它首先针对基目录解析;在这种情况下，此函数有效地充当 $$clean_path （） 。filePath

此功能在Qt 5.0中引入。

另请参阅absolute_path （） ， clean_path （） ）。

## 替换（字符串、old_string、new_string）

替换提供为 的变量内容中的每个实例。例如，代码old_stringnew_stringstring

```
MESSAGE = This is a tent.
message($$replace(MESSAGE, tent, test))
```

打印消息：

```
This is a test.
```

## resolve_depends（变量名、前缀）

这是您通常不需要的内部函数。

此功能在Qt 5.0中引入。

## 反向（变量名）

以相反的顺序返回 的值。variablename

此功能在Qt 5.0中引入。

## 部分（变量名、分隔符、开始、结束）

返回 的值的一部分。此函数是QString： ： section 的包装器。variablename

例如，以下调用输出：surname

```
CONTACT = firstname:middlename:surname:phone
message($$section(CONTACT, :, 2, 2))
```

## 阴影（路径）

此功能在Qt 5.0中引入。

### shell_path（路径）

将所有目录分隔符转换为与生成项目时使用的 shell（即 make 工具调用的 shell）兼容的分隔符。例如，当使用 Windows 外壳时，斜杠将转换为反斜杠。path

此功能在Qt 5.0中引入。

另请参阅system_path（）。

### shell_quote（参数）

引号用于生成项目时使用的 shell。arg

此功能在Qt 5.0中引入。

另请参阅system_quote（）。

### 大小（变量名）

返回 的值的个数。variablename

另请参阅str_size（）。

### sort_depends（变量名、前缀）

这是您通常不需要的内部函数。

此功能在Qt 5.0中引入。

### 已排序（变量名）

返回按 ASCII 升序排序的条目中的值列表。variablename

数字排序可以通过在format_number（） 函数的帮助下将值零填充到固定长度来完成。

此功能在Qt 5.8中引入。

### 拆分（变量名、分隔符）

将 的值拆分为单独的值，并将它们作为列表返回。此函数是QString：：split 的包装器。variablename

例如：

```
CONTACT = firstname:middlename:surname:phone
message($$split(CONTACT, :))
```

### sprintf（string， arguments...）

将 %1-%9 替换为在逗号分隔的函数列表中传递的参数，并返回已处理的字符串。stringarguments

此函数可用于实现许多常见的字符串切片操作：

```
# $$left(VAR, len)
left = $$str_member(VAR, 0, $$num_add($$len, -1))

# $$right(VAR, len)
right = $$str_member(VAR, -$$num, -1)

# $$mid(VAR, off, len)
mid = $$str_member(VAR, $$off, $$num_add($$off, $$len, -1))

# $$mid(VAR, off)
mid = $$str_member(VAR, $$off, -1)

# $$reverse(VAR)
reverse = $$str_member(VAR, -1, 0)
```

**注意：**在这些实现中，零参数需要单独处理。len

另请参阅成员（），num_add（）。

此功能在Qt 5.8中引入。

## str_size（参数）

返回参数中的字符数。

另请参阅大小（）。

此功能在Qt 5.8中引入。

## system（command[， mode[， stsvar]]）

您可以使用函数的此变体从命令中获取 stdout 并将其分配给变量。system

例如：



```
UNAME = $$system(uname -s)
contains( UNAME, [lL]inux ):message( This looks like Linux ($$UNAME) to me )
```

像$$cat（）一样，*模式*参数采用,,,andas值。但是，遗留的单词拆分规则（即空或和）略有不同。bloblinestruefalsetruefalse

如果传递，命令的退出状态将存储在该变量中。如果命令崩溃，则状态将为 -1，否则为命令选择的非负退出代码。通常，将状态与零（成功）进行比较就足够了。stsvar

另请参阅system（） 的测试变体。

将所有目录分隔符转换为与函数用于调用命令的外壳兼容的分隔符。例如，斜杠将转换为 Windows 外壳的反斜杠。pathsystem()

此功能在Qt 5.0中引入。

另请参阅shell_path ()。

### system_quote（参数）

引号对于函数使用的外壳。argsystem()

此功能在Qt 5.0中引入。

另请参阅shell_quote ()。

### take_first（变量名）

返回 的第一个值并将其从源变量中删除。variablename

例如，这为实现队列提供了便利。

此功能在Qt 5.8中引入。

另请参阅take_last ()，first ()。

### take_last（变量名）

返回 的最后一个值并将其从源变量中删除。variablename

例如，这为实现堆栈提供了便利。

此功能在Qt 5.8中引入。

另请参阅take_first ()，last ()。

### 唯一（变量名）

返回已删除重复条目的值列表。例如：variablename

```
ARGS = 1 2 3 2 5 1
ARGS = $$unique(ARGS) #1 2 3 5
```

### 上层（arg1 [， arg2 ...， argn]）

采用任意数量的参数并将其转换为大写。

另请参阅下 ()。

### val_escape（变量名）

以允许将其解析为 qmake 代码的方式转义 的值。variablename

此功能在Qt 5.0中引入。


联系我们

**公司**

关于我们
投资者
编辑部
职业
办公地点

**发 牌**

条款和条件
开源
常见问题

**支持**

支持服务
专业服务
合作 伙伴
训练

**对于客户**

支持中心
下载
Qt登录
联系我们
客户成功案例

**社区**

为Qt做贡献
论坛
维基
下载
市场



反馈 登录