

## ►vJass 系列教程 3

## debug 预处理符与地图调试

Aeris ▶ NJU ▶ 2008-6-30

[illegible]

## vJass 系列教程 3

### debug 预处理符与地图调试

#### 本教程内容简介

在上一章教程里，我向大家介绍了 **vjass** 的全局变量自由定义机制。这一章，将向大家介绍 **debug** 预处理符。这个语法特征虽然很简单，却非常实用，特别是在地图的调试中，正如其名。

#### 语法概述

在暴雪的 **Jass** 里，有个不起眼的关键字：**debug**。多年来一直没有多少人关注它，因为它仅仅是空有其名而没有任何实际用处。不少人推测可能是暴雪内部使用的，而在发行的魔兽中，这个关键字已经没有任何效果了。一个语言的关键字就那么几个，放着不用是浪费。因此，**vjass** 的作者决定把这个尘封了多年的关键字重新启用，赋予它新的意义。

#### 语法说明

**vjass** 的编译器在处理代码之前有个**预处理**的过程，这个过程仅仅是作一些文本替换工作，如同 **C/C++** 的预处理器一样。**debug** 关键字是预处理符，它在预处理的过程中，控制文本的生成与否。

**debug** 预处理符的语法很简单，如下所示：

#### debug vJass 语句

可以在任意一行 **vjass** 之前加上 **debug** 关键字，例如：

```
// 调试模式下检测某标志
debug if (flag == false) then
    debug call BJDebugMsg("Error occurred in function xxxx, .....")
debug endif
```

**debug** 预处理符的处理机制是：当编译器处于 **debug 模式** 时（NewGen 的 JassHelper 菜单里第 2 项），预处理时将简单去掉 **debug** 关键字，保留其后的语句；当编译器处于 **release 模式** 时，预处理时会将以 **debug** 开头的行全部移除。可见，**debug** 预处理符的作用是：引入一些只在 **debug 模式** 下起作用，而在 **release 模式** 下无效的语句，主要用于地图的调试。

#### 例子

这是一个简单的例子（假设 **flag** 是某操作是否成功的标志）

```
if (flag == false) then
    // 出错
    debug call BJDebugMsg("Error occurred in function xxxx, .....")
else
    // 系统正常
    ...
```

**endif**

这个例子在 debug 模式下，会被编译成：

```
if (flag == false) then
    // 出错
    call BJDebugMsg("Error occurred in function xxxx, .....")
else
    // 系统正常
    ...
endif
```

在 release 模式下，会被编译成

```
if (flag == false) then
    // 出错
    // debug call BJDebugMsg("Error occurred in function xxxx, .....")
else
    // 系统正常
    ...
endif
```

可以看到，debug 模式下，debug 关键字被去除了，这一行语句发挥了作用；而在 release 模式下，整行都被注释掉了，失去了作用，因此，这个程序在 debug 模式下出错时，会打印出出错信息，而在 release 模式下出错时，将什么都不会显示。

#### debug 预处理符与地图调试

魔兽地图的调试没有什么特别方便的工具，虽然 NewGen 自带了调试器，但是传统的“打印式调试”仍然是主要调试手段。魔兽的打印式调试一般是借助暴雪提供的 BJ 函数 BJDebugMsg（内部使用 DisplayTimedTextToPlayer 函数实现）向屏幕上输出调试信息，比如某某变量的值啊等等的。打印式调试是经典而又有效的调试方法。但在魔兽中使用的时候，大量调试语句的清除是一个问题，因为在地图发行时，调试语句是要清除的，不能给玩家看太多的调试信息，否则因为大量无用信息充斥屏幕，容易让玩家失去兴致。大部分地图作者选择手工慢慢清除调试语句，费时费力又容易出错。debug 预处理符正好可以帮上忙。把调试语句前面加上 debug 预处理符，在调试模式下可以输出调试信息。地图完成之后发布时，切换到 release 模式，所有的调试语句都会被自动去除。