BlueantStudio

QQ 群：325880743，微信公众号：blueantstudio

http://www.blueantstudio.net

Script.NET V4

[SCRIPT.NET 调试器设计说明]

Script.NET 架构设计文档

目录

Script.NET 调试器设计说明

1. 调试器

1.1. 脚本调试的流程和思路

脚本调试的一般思路是对被调试脚本打桩，也就是在被调试脚本的每一个有效行前面增加一个断点桩，对于目前 Tcl 脚本的调试，是将被调试脚本加载到内存中逐句的分析，在每个有效代码行前面插入了一个扩展命令，命令格式是：

td lineno;

其中 td 是自己扩展的调试扩展命令，lineno 是实际的代码行号。

分析完之后，在内存中运行修改过的脚本，这样脚本执行到每一行的时候都会先执行 td 扩展命令，并将这一行的行号传递给这个命令。

在 td 扩展命令中，会判断当前的调试状态，如果是单步，则每次都暂停下来等候用户的操作，如果是断点方式，则判断当前行是否在断点列表中，如果不在就直接退出 td 命令，如果在就暂停下来。

对于单步又分为单步向下，单步进入子函数，单步退出子函数三种类型，对于进入和退出子函数是通过对当前堆栈深度的判断来判断的，例如当前深度为 2，如果下一次深度为 3，说明进入了一个子函数，如果当前操作是单步进入子函数，则就需要停下来，否则就不停下来。

对于 Python 的调试，思路虽然类似，但因为 Python 自己已经提供了一个调试扩展包，叫 pdb，因此只要按照 pdb 的要求，从 pdb 派生一个 python 调试类，通过派生的调试类来执行脚本就可以实现调试，同时断点是依靠在派生类的一些函数中调用解释器中的扩展命令来实现的，总体来说 python 的调试实现会简单一些，而且因为是语言内建的，不需要自己解析脚本文件，因此通用性更好一些。Ruby 可能也有类似的实现方法。

解释器组件中的 SetDebugEvent 函数是一个对调试来说比较关键的函数，这个函数的触发是编辑器中点击单步、单步进入等操作按钮的时候触发的，对每一种操作都有标准的命令码定义，只要在这个函数中做相应的调试操作处理就可以。

InitialDebugMode 函数用于初始化调试模式。主要是初始化调试窗口等操作。

RefreshDebugWindow 函数用于刷新调试窗口，支持的 4 个标准调试窗口是变量、对象、过程、堆栈窗口。

1.2. 调试事件说明

调试器事件说明，定义在 IInterp.h 中。分为调试器事件和平台事件，平台事件是由平台某个

功能菜单触发的，发送给调试器，调试器根据事件类型进行相应处理。调试器事件是调试器执行的时候发送给外部其他模块的事件，例如设置当前行是发送给编辑器模块的回调事件。脚本调试的时候是由编辑器启动一个独立的调试线程来调用解释器的 RunScriptFile 函数的，而界面上的操作（例如调试过程中进行的单步操作或显示变量窗口操作）是在平台界面线程中的。大部分脚本语言对于线程之间的操作不会有什么问题，但有些脚本就有问题，例如 ruby 脚本要求当前调用 ruby API 的线程是 ruby 脚本运行时候的原始线程，如果不是就可能异常，因此 ruby 脚本调试时候如果直接点击界面操作的刷新变量窗口，就会导致异常，解决方法就是调试过程中所有需要操作 ruby API 的操作都通过调试事件，激活调试过程中的 DebugBreak 函数中的调试等待事件，对操作逐个进行判断，然后在调试线程中执行相应的操作。为此，平台调试命令事件增加了一些新的事件。

```
// 调试事件类型
enum{
    // 以下为调试器回调事件
    IDB_USER_CALL,          // 调用函数
    IDB_USER_LINE,          // 执行一行
    IDB_USER_RETURN,        // 函数返回
    IDB_USER_EXCEPTION,     // 发生异常
    IDB_USER_SETCURRENTLINE,// 设置当前行

    // 以下为平台调试命令事件
    IDB_STOP= 10,           // 终止
    IDB_NEXT,               // 执行到下一个断点
    IDB_GOTO,               // 执行到某个行断点或命名断点
    IDB_STEPINTO,           // 单步进入函数
    IDB_SETPOUT,            // 从函数返回
    IDB_STEP,               // 单步不进入函数
    IDB_END,                // 通知解释器文件调试结束，可以执行善后处理

    IDB_SETFILE = 20,       // 设置被调试的文件

    IDB_REFRESH_DEBUGWINDOW = 30,   // 刷新调试窗口
```

```
};
```