

在 Windows 下使用 Notepad++ 和 xdebug 调试 php 脚本

- mikespook
- mikespook@gmail.com
- <http://www.mikespook.com>
- 08年3月3日

介绍

Notepad++ 是开放源代码的可替代记事本的编辑器。它运行于 MS Windows 环境，支持多种编程语言。可以浏览 <http://notepad-plus.sourceforge.net/> 了解更多相关信息。

Xdebug 是 php 的一个扩展，它提供了对 php 脚本进行除错、追踪、检查的各种功能。可以浏览 <http://xdebug.org> 了解更多相关信息。

下载

Notepad++ 下载地址: <http://nchc.dl.sourceforge.net/sourceforge/notepad-plus/npp.4.7.5.Installer.exe>

Notepad++ 插件 DBGP plugin: http://nchc.dl.sourceforge.net/sourceforge/npp-plugins/DBGpPlugin_0_7b_fix_dll.zip

xdebug: <http://xdebug.org/link.php?url=xdebug202-52-win>

* xdebug 的版本需要匹配环境使用的 PHP 版本，这里提供的链接为 PHP 5.2.1-5.2.7 使用 xdebug。

安装

Notepad++ 的安装方式如同大多数 Windows 下软件一样是非常简单的。只需按照屏幕提示设置并点击“下一步”，最后“完成”即可安装成功。

为了让 Notepad++ 和 xdebug 协同工作，需要安装 DBGP plugin。下载后只需解压缩至 Notepad++ 安装目录中的 plugins 目录即可，如：C:\Program Files\Notepad++\plugins。

xdebug 的安装请参考：<http://www.mikespook.com/index.php/archives/34>。需要说明的是，如果同时安装有 xcache 一定要先加载 xcache，再加载 xdebug。否则会造成 php 运行异常。

使用

首先，将编写分别使用递归方式和使用循环方式计算 100 的阶乘的两个函数，如下：

```
<?php
function fl($x)
// 循环计算 $x 的阶乘
```

```
{  
  
    for($i = $x - 1; $i > 1; $i--)  
    {  
        $x *= $i;  
    }  
    return $x;  
}
```

```
function f2($x)  
// 递归计算 $x 的阶乘  
{  
    if($x == 1)  
    {  
        return $x;  
    }  
    else  
    {  
        $y = $x - 1;  
        return $x * f2($y);  
    }  
}
```

```
echo 'f1: ' . f1(10);  
echo '<br />';  
echo 'f2: ' . f2(10);
```

将该文件放入可通过 web 访问并由 php 正确解析的目录，访问结果如图：

```
f1: 3628800  
f2: 3628800
```

使用 Notepad++ 打开这个文件。通常在文件上点击鼠标右键（或者左键，如果你跟我一个方向的话），选择“Edit with Notepad++”，如图：

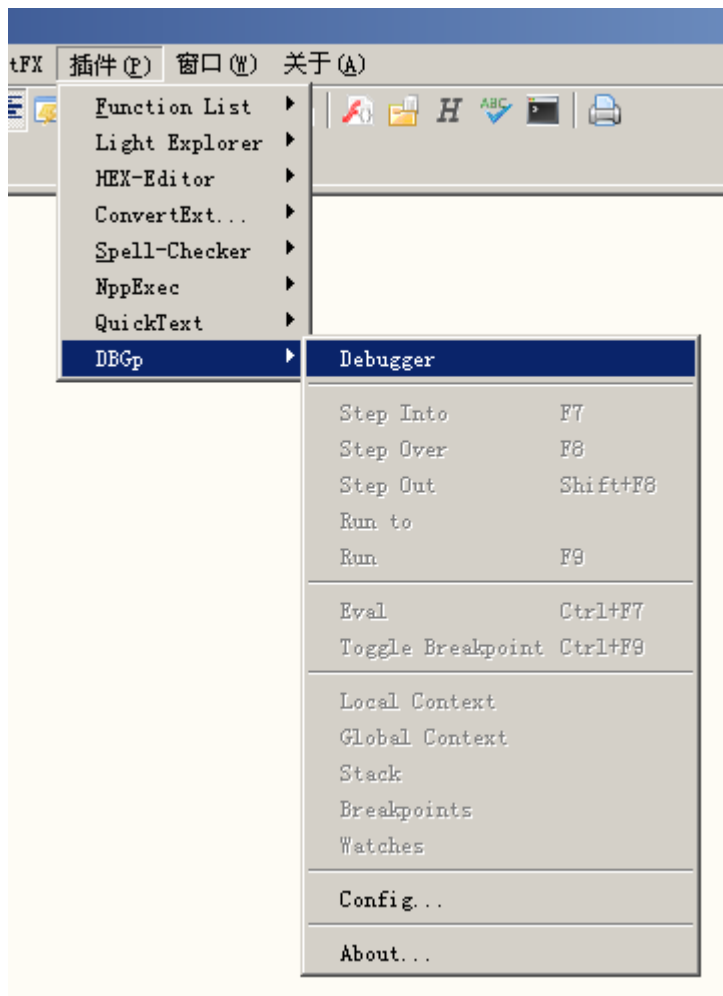


如果你没有这项，或者无法使用 Notepad++，请检查是否正确的安装了 Notepad++。打开后的界面如图：

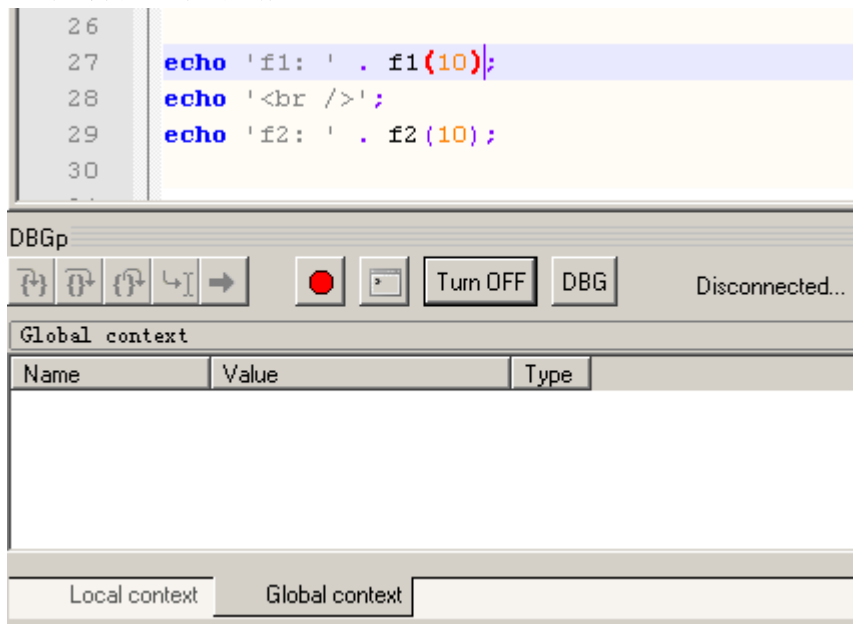
A screenshot of the Notepad++ application window. The title bar reads 'Notepad++ - F:\www\foobar.php'. The menu bar includes '文件 (F)', '编辑 (E)', '搜索 (S)', '视图 (V)', '格式 (M)', '语言 (L)', and '设置 (T)'. The toolbar contains various icons for file operations and editing. The main text area shows PHP code for calculating factorials using two functions: f1 (iterative) and f2 (recursive). The code is as follows:

```
1 <?php
2 function f1($x)
3 // 循环计算 $x 的阶乘
4 {
5
6     for($i = $x - 1; $i > 1; $i--)
7     {
8         $x *= $i;
9     }
10    return $x;
11 }
12
13 function f2($x)
14 // 递归计算 $x 的阶乘
15 {
16     if($x == 1)
17     {
18         return $x;
19     }
20     else
21     {
22         $y = $x - 1;
23         return $x * f2($y);
24     }
25 }
26
27 echo 'f1: ' . f1(10);
28 echo '<br />';
29 echo 'f2: ' . f2(10);
30
31
```

在 Notepad++ 的菜单“插件”中选择“DBGp”并点击“Debug”：



会在编辑器下半部分打开调试窗口：



此时就已经打开了编辑器端的调试功能。

使用 `XDEBUG_SESSION_START=1` 参数访问刚才编写的文件，如：
http://127.0.0.1/foobar.php?XDEBUG_SESSION_START=1

此时会发现 Notepad++ 窗口会闪烁一下，这说明 Notepad++ 的 DBGp 插件已经生效。将光标放置到需要设置断点的位置，并点击有红点的按钮，光标所在行前会显示一个红点，表示断点行，如图：

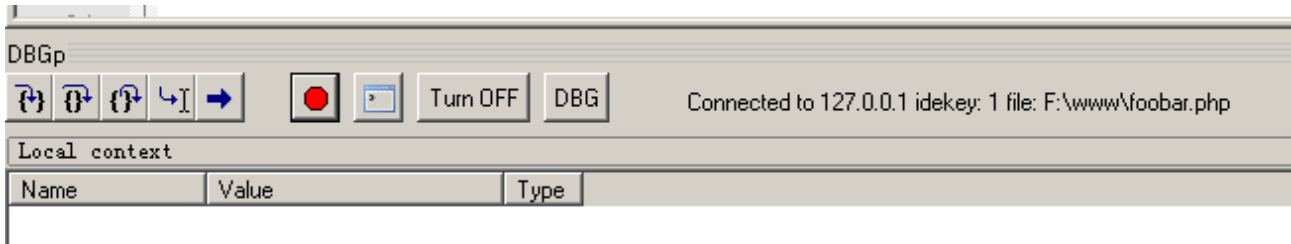
```
6         for($i = $x - 1; $i > 1; $i--)
7         {
8             $x *= $i;
9         }
```

此时访问/foobar.php?XDEBUG_SESSION_START=1，会发现浏览器无法正确显示出页面。其实这是由于 Notepad++ 已经将 PHP 运行中断在断点处，等待调试：

```
6         for($i = $x - 1; $i > 1; $i--)
7         {
8             $x *= $i;
9         }
```

Breakpoints			
	Type	Breakpoint data	Hits
●	Line	F:\www\foobar.php:8	1

之前无法使用的按钮，现在也可点击，可实现“Step Into”、“Step Over”、“Step Out”、“Run to cursor”、“Run”操作：



打开 Local Context 标签，点击鼠标右键选择“Refresh”，可以显示当前所有局部变量：

Local context		
Name	Value	Type
\$i	7	int
\$x	720	int

Local context Global context

而对 Global Context 标签做同样的操作将显示所有全局变量：


Global context		
Name	Value	Type
\$_COOKIE		array
\$_ENV		array
\$_FILES		array
\$_GET		array
\$_POST		array

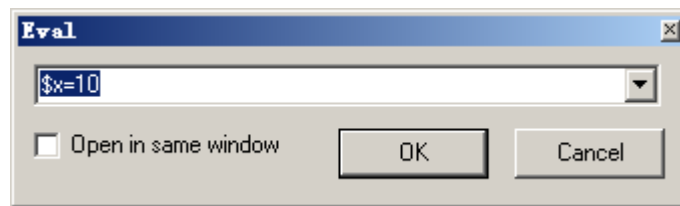
Local context Global context

Stack 标签可以显示栈的使用情况，这对于调试函数调用很有帮助。例如本例中的递归方式计算阶乘：

Stack				
Level	File	Line	Where	Type
0	F:\www\foobar.php	16	f2	file
1	F:\www\foobar.php	23	f2	file
2	F:\www\foobar.php	23	f2	file
3	F:\www\foobar.php	23	f2	file
4	F:\www\foobar.php	23	f2	file

调试完毕或后可以将光标放置在断点处，再次点击断点按钮取消断点。然后点击“Run”按钮或按 F9 继续程序的执行。也可直接点击“Trun OFF”按钮关闭调试功能。

特别要提一下的是 Eval 按钮  的功能。这个按钮可以动态改变某个变量的内容，以方便调试：



Eval		
Name	Value	Type
... \$x=10	10	int

Local context		
Name	Value	Type
\$y		uni..
\$x	10	int

至此，Notepad++ 配合 xdebug 对 PHP 进行调试的基本功能就介绍完了。调试是非常讲究技巧和经验的，真正发挥调试的威力，还要大家在实际使用中总结和提高。在我有时间的时候，我会继续 VIM 配合 xdebug 的介绍，同时录制一些屏幕录像给大家参考。