**明源性能测试学习及指引**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **作者** | 赵思杰 | **当前版本号** | V1.1 | **使用范围** | ● 研发 ○ 明源内部 ○ 客户 |

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| 修订日期 | 版本号 | 修订说明 | 修订人 |
| 2012.01.30 | V1.1 | 1、修改URL-based scrip脚本录制的描述  2、修改数据缓存与计划缓存的描述 | 赵思杰 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 明源性能测试概况 4](#_Toc315683543)

[1.1 名词解释 4](#_Toc315683544)

[1.2 明源性能测试的目的 4](#_Toc315683545)

[1.3明源性能测试流程 5](#_Toc315683546)

[1.4明源性能测试工具 6](#_Toc315683547)

[2 HttpWatch介绍 6](#_Toc315683548)

[2.1工具简介 6](#_Toc315683549)

[2.2工具使用 6](#_Toc315683550)

[3 DynaTrace介绍 7](#_Toc315683551)

[3.1 工具简介 7](#_Toc315683552)

[3.2 工具使用 7](#_Toc315683553)

[4 LoadRunner介绍 10](#_Toc315683554)

[4.1 工具简介 10](#_Toc315683555)

[4.2 相关组件 10](#_Toc315683556)

[4.3 工具使用 10](#_Toc315683557)

[4.3.1 VuGen的使用 10](#_Toc315683558)

[4.3.2 Controller使用 36](#_Toc315683559)

[4.3.3 Load Generator使用 40](#_Toc315683560)

[4.3.4 场景运行 45](#_Toc315683561)

[4.3.5 Analysis使用 49](#_Toc315683562)

[4.3.6 IP欺骗 52](#_Toc315683563)

[5 性能相关知识学习 57](#_Toc315683564)

[5.1 交换机限速 57](#_Toc315683565)

[5.2 WEB服务器与负载均衡 58](#_Toc315683566)

[5.3 DB快照与SQL索引优化 59](#_Toc315683567)

[5.4 数据缓存与计划缓存 59](#_Toc315683568)

# 1 明源性能测试概况

## 1.1 名词解释

* 大数据量测试：将数据库翻倍到需求要求的数量级后，对大数据量需求中要求的测试点逐个进行点击的性能测试
* 单交易并发测试：将数据库翻倍到需求要求的数量级后，对需求中要求测试的场景进行单个场景的多用户并发的性能测试
* 组合场景测试：将数据库翻倍到需求要求的数量级后，将需求要求进行组合测试的单场景进行组合后的多用户测试
* 稳定性场景测试：将组合场景运行24小时

大数据量、单交易、组合场景三者之间的异同：

* 在数据库上:三种测试是用的相同数据的库，所以数据库中的数据在满足需求的同时，也要满足三种测试的数据量
* 在用户数上：大数据量是单个用户的测试；单交易并发是多个用户的强并发测试；组合场景是多个用户1小时内用户操作，属弱并发
* 在场景上：大数据量是测试人员直接进行人工点击的测试；单交易并发是用工具模拟一个场景的测试；组合场景是用工具模拟多个场景组合在一起的测试

## 1.2 明源性能测试的目的

目前我们主要是通过大数据量测试、单交易并发测试、组合场景测试发现系统中存在的性能不达标点，以提供给开发进行调优

## 1.3明源性能测试流程

* 1. 性能测试需求编写
  2. 性能测试方案编写
  3. 性能测试数据翻倍
  4. 性能测试用例编写
  5. 性能测试执行
  6. 性能调优
  7. 性能回归测试

**性能测试需求编写**：目前性能测试需求主要是产品经理进行编写，性能测试人员需对需求进行评审，以评审需求的合理性，合理性主要是性能测试所要求的数据量是否合理、测试场景是否合理、数据翻倍方案是否合理。我们还要关心数据量的来源，也就是需求所要求的数据量是如何推导出来的，如ERP3.0的数据量的来源就是以万科2010年的销售为基础进行的推导。

**性能测试方案的编写**：目前性能测试方案由性能测试Leader在写，方案主要需明确测试的环境、测试的策略和测试的风险。环境包括：WEB服务器的软硬件环境，DB服务器的软硬件环境，客户端的软硬件环境，网络环境等；测试的策略包括测试的类型（大数据量、单交易并发、组合场景），各种类型对应的方法和工具脚本设置，测试执行和监控策略等。

**性能测试数据翻倍**：目前这个由开发在做，但我们测试人员需要检查，翻倍的数据需满足大数据量测试、单并发测试和组合场景测试数据要求。

**性能测试用例编写**：由性能测试人员进行编写，用例需要明确并发数、测试数据准备、场景的设置、场景的每一步操作、操作对应的事务以及需要测试的功能点的性能达标时间等。

**性能测试执行：**包括大数据量的执行、脚本的参数化SQL编写、脚本的录制与编辑、脚本的执行、场景的监控等。

**性能调优：**由开发人员进行调优。

**性能回归测试：**调优后需再次进行测试执行。

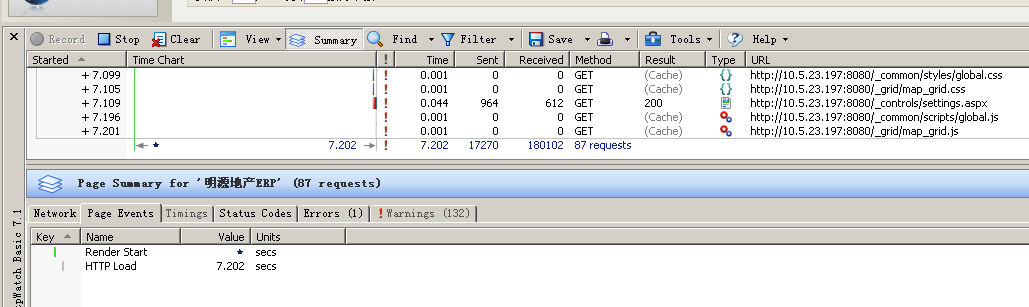
## 1.4明源性能测试工具

目前我们用到了秒表、HttpWatch、DynaTrace、LoadRunner等，其中秒表是开发提供的一个小工具，可以直接进行目测的性能测试，主要用于大数据量点击测试中。

# 2 HttpWatch介绍

## 2.1工具简介

HttpWatch是强大的网页数据分析工具，集成在Internet Explorer工具栏，.包括网页摘要、Cookies管理、缓存管理、消息头发送/接受、字符查询、POST 数据和目录管理功能、报告输出等

如图：

## 2.2工具使用



大数据量测试我们用秒表外，HttpWatch也是我们常用的工具，一般情况下，我们先用秒表测试，发现性能不稳定点，或性能测试不达标点，我们需用到HttpWatch。典型案例：HttpWatch+任务管理器的进程，发现过内存泄露问题，销售系统中集中开盘的选房确认，每切换一次房间时会产生3M的内存占用，导致选房房间展示时间一次比一次长，HttpWatch可以很清楚的显示每次房间展示的时间。

# 3 DynaTrace介绍

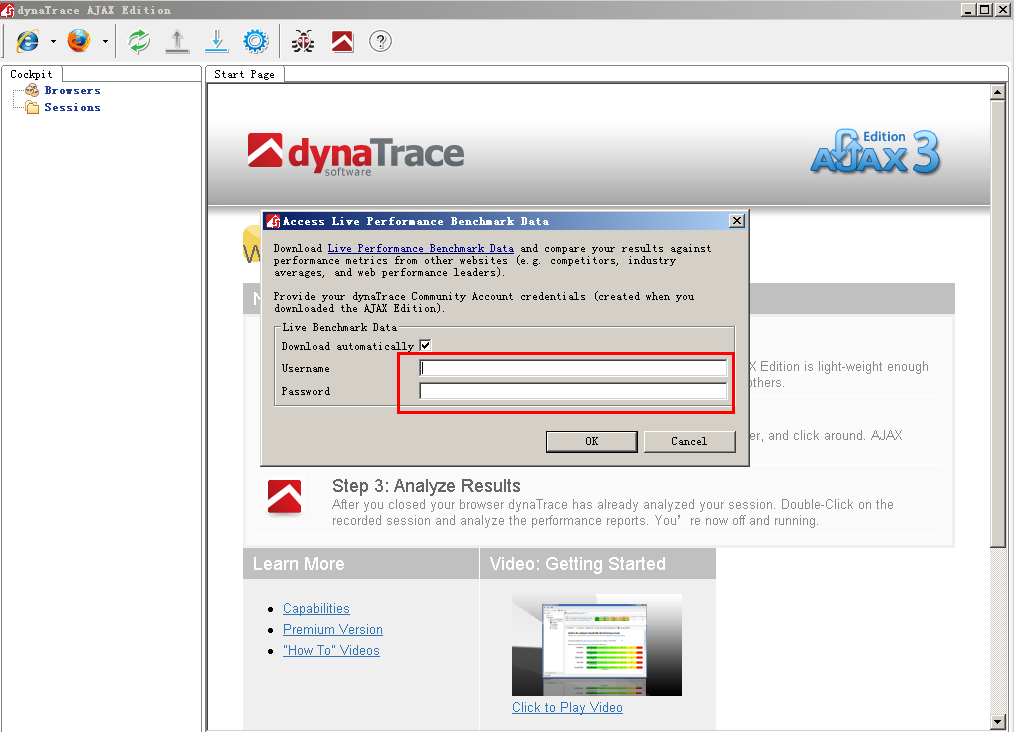
## 3.1 工具简介

dynaTrace Ajax是一个详细的底层追踪工具，它不仅可以显示所有请求和文件在网络中传输的时间，还会记录浏览器Render、CPU消耗、JavaScript解析和运行情况等详细的信息

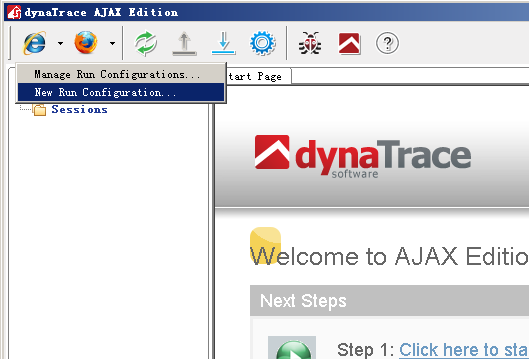
## 3.2 工具使用

一、安装dynatrace-AJAX-edition-3.1.0.763，直接下一步、下一步就安装OK了。

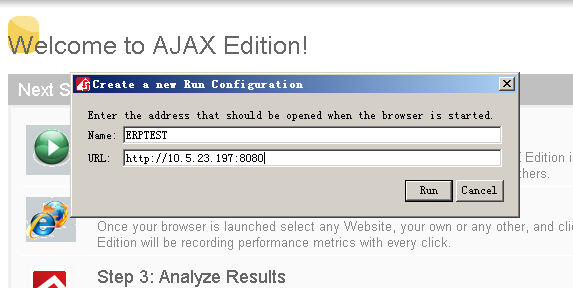
二、安装完成后，进入如下图所示界面，注册信息为：Username：[ptemail@126.com](mailto:ptemail@126.com) Password：CK6h42（区分大小写）。



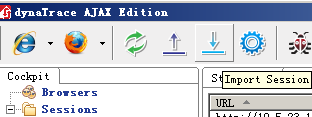
三、然后选择你系统对应IE的下拉框，在下拉框中选择”New Run Configuration…”即可



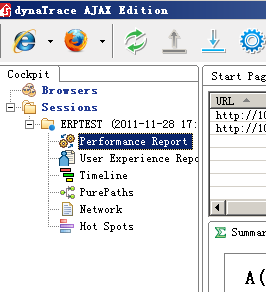
四、然后在弹出的框中输入任意名称，再输入你要测试的HTTP站点，点击即可RUN，你就可以操作了系统进行测试了



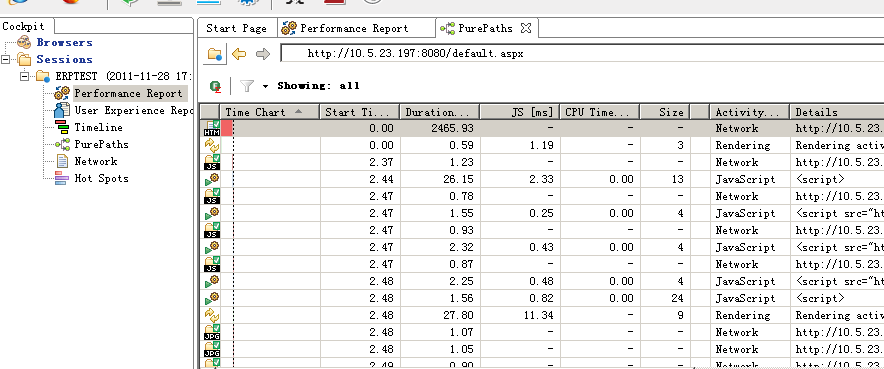
五、测试完成后，关闭当前页面，就可以进行保存了



六、选择”Performance Report”就可以查看录制到操作的URL页面



七、双击右边中对应的URL链接，即可以查看到该页面的详细性能信息了



该工具我们使用的较少，该工具更适合开发使用。我们的ERP系统使用该工具进行监控时，ERP的某些功能点无法显示或操作，在使用时我们需要禁用IE的调试，如果开启了IE调试，会经常弹出调试的界面。HttpWatch无法监控时，我们需要使用该工具，比如JS运行较长时间时。例如销售系统的集中开盘的摇号功能就非常适合用该工具进行监控

# 4 LoadRunner介绍

## 4.1 工具简介

LoadRunner，是一种预测系统行为和性能的负载测试工具。通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，LoadRunner能够对整个企业架构进行测试。通过使用 LoadRunner，企业能最大限度地缩短测试时间，优化性能和加速应用系统的发布周期。 LoadRunner是一种适用于各种体系架构的自动负载测试工具，它能预测系统行为并优化系统性能。

## 4.2 相关组件

* **VuGen（虚拟用户生成器）** 用于捕获最终用户业务流程和创建自动性能测试脚本 （也称为虚拟用户脚本）。
* **Controller （控制器）**用于组织、驱动、管理和监控负载测试。
* **Load Generator（负载生成器）**用于通过运行虚拟用户生成负载。
* **Analysis （分析器）**查看、分析和比较性能结果。

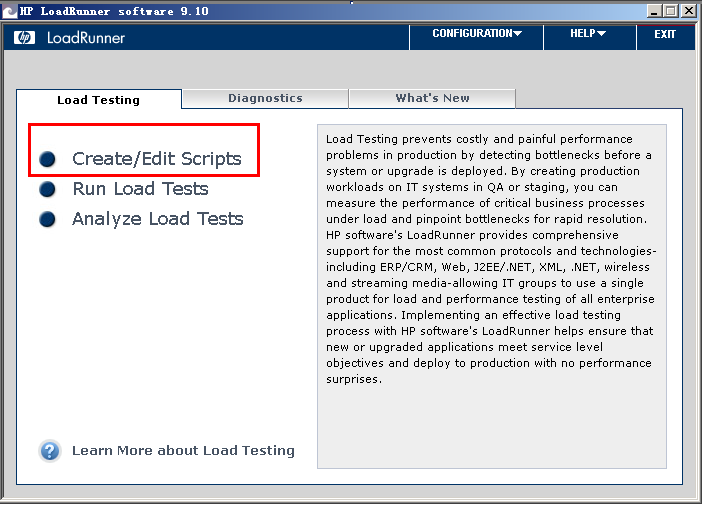
## 4.3 工具使用

### 4.3.1 VuGen的使用

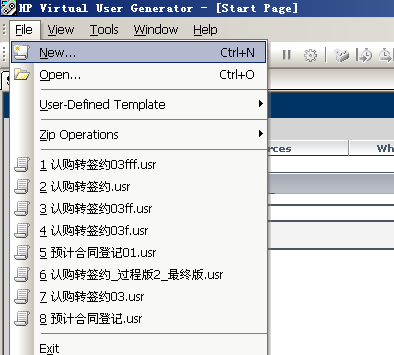
Virtual User Generator 是一种基于录制回放的工具，当你按照业务流程执行了某个软件，它会将你的操作中产生的协议录下来，自动转化成脚本，执行完成对用户行为的模拟，从而进一步对系统产生的负载。而性能测试的第一步也是最重要的一步就是生成虚拟用户脚本。下面我们就来开始VuGen的学习

#### 4.3.1.1 脚本开始录制

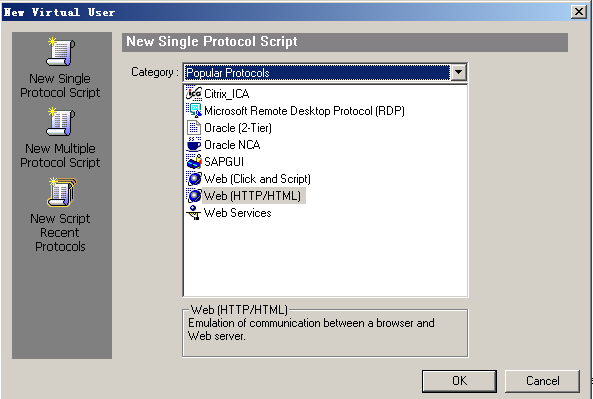
一、找到LoadRunner，双击打开，然后在主界面上选择Create/ Edit Scripts



二、在弹出的VUGen界面中，选择”File”下的”New”



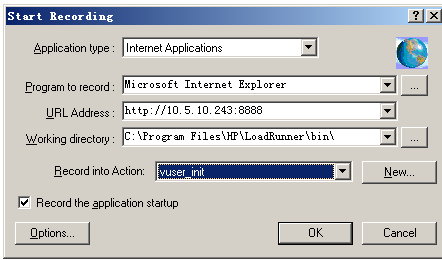
三、在弹出的New Virtual User界面，选择”New Single Protocol Script”中的Web(HTTP/HTML)



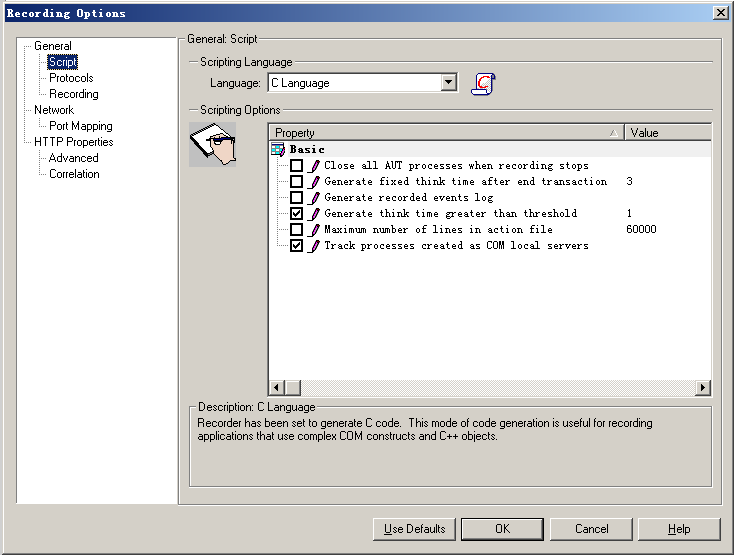
* New Single Protocol Script ：设置单协议
* New Multiple Protocol Script：设置多协议
* New Script Recent Protocols：选择之前用过的协议

我们的ERP目前用Web(HTTP/HTML)协议录制足够

四、点击OK,弹出如下界面

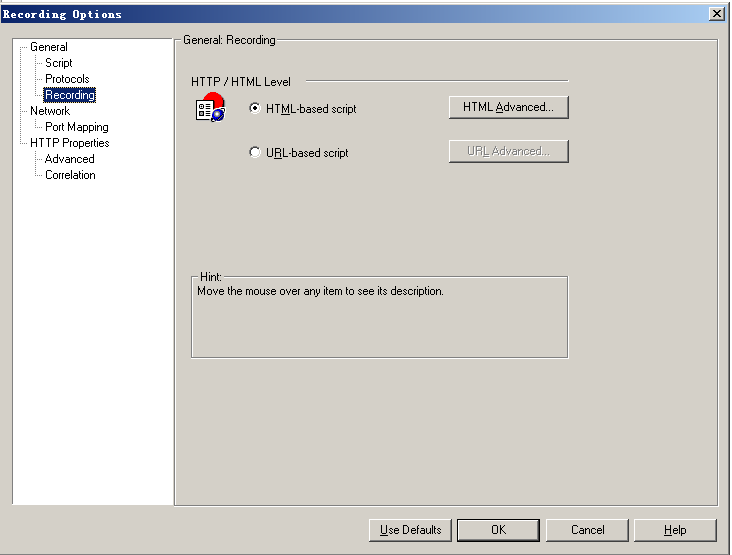


* Application type：应用程序类型，我们当然选internet Applications
* Program to record：录制的程序，我们当然是用IE录制了。这里可以选择IE，LoadRunner 9.1版对IE7支持最好，所以这里我们选择IE7所在的目录文件
* URL Address：URL地址，就是我们要录制的程序的URL地址
* Working directory:工作目录，这个就是LoadRunner的目录，我们默认
* Record into Action:是我们录制从哪开始。有Vuser\_init、Action、Vuser\_end三个选择，一般情况下，我们把打开网站和登录放在Vuser\_init里面，所以我们这里选择Vuser\_init
* Options… 如下图

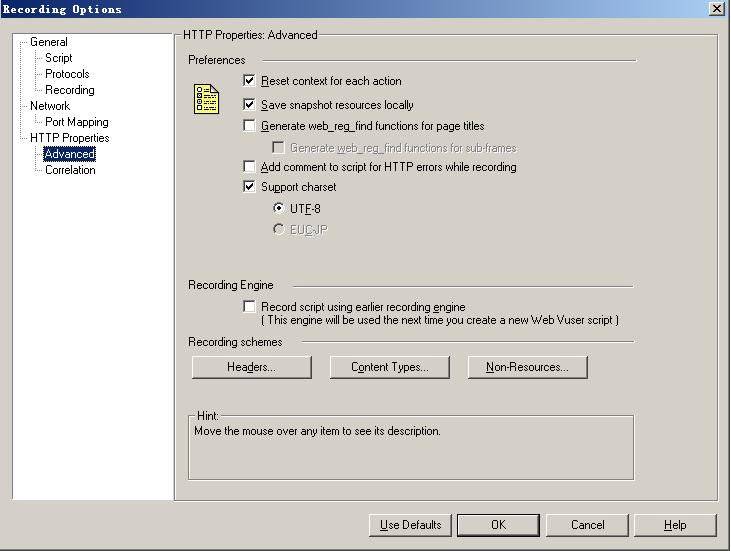


这里我们只讲重点，下去可查资料对相关设置进行了解

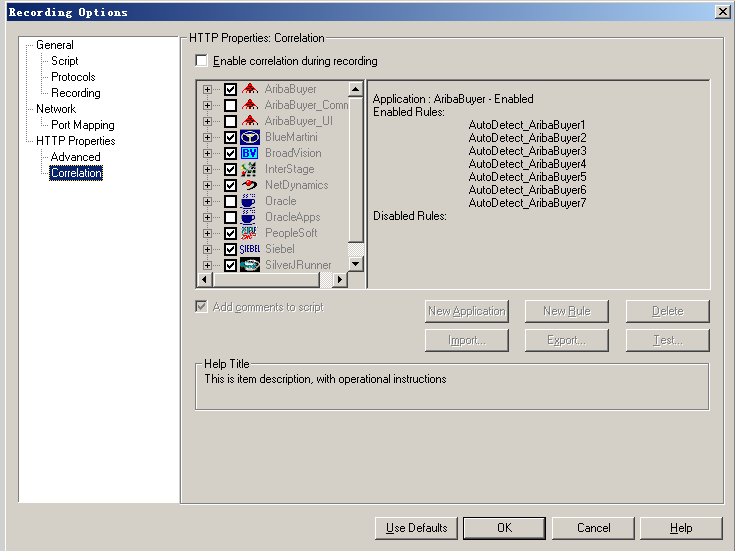
首先来看General-Recording，里面有两个重要的选择HTML-based script和URL-based script。HTML-base script：指以HTML为单元进行脚本录制，为用户的每个HTML操作生成单独的步骤，这种脚本看上去比较直观。URL-based script：可以捕获所有作为用户操作结果而发送到服务器的HTTP请求，然后为用户的每个请求分别生成对应方法。目前我们大多采用HTML-based script，不排除个别地方用HTML-based script录制的无法回放成功。目前我录了三十多个脚本，只碰到过两个地方用HTML-based script录制无法回放成功的，就是电子开盘1.0中的转预约排号和快速认购，这时我们可以采用URL-based script进行录制。（目前我发现我们的系统中，用HTML-based script录制时，当要使用关联的GUID在使用点的HTML处返回或使用点的HTML之后返回，我们关联无法使用时，需采用URL-based script，用URL-based script录制后，可以发现需要使用的GUID在使用之前的URL就返回了。至于还有没有其它地方也需要用到URL-based script，待将来总结）



接着我们来看HTTP Properties下的Advanced设置，我们目前只关注Support charset选项，这里用来选择我们录制脚本时，采用的编码方式，这里我们选的是UTF-8，虽然选这种编码还是会出现乱码，但后面会提到如何去解决这种问题

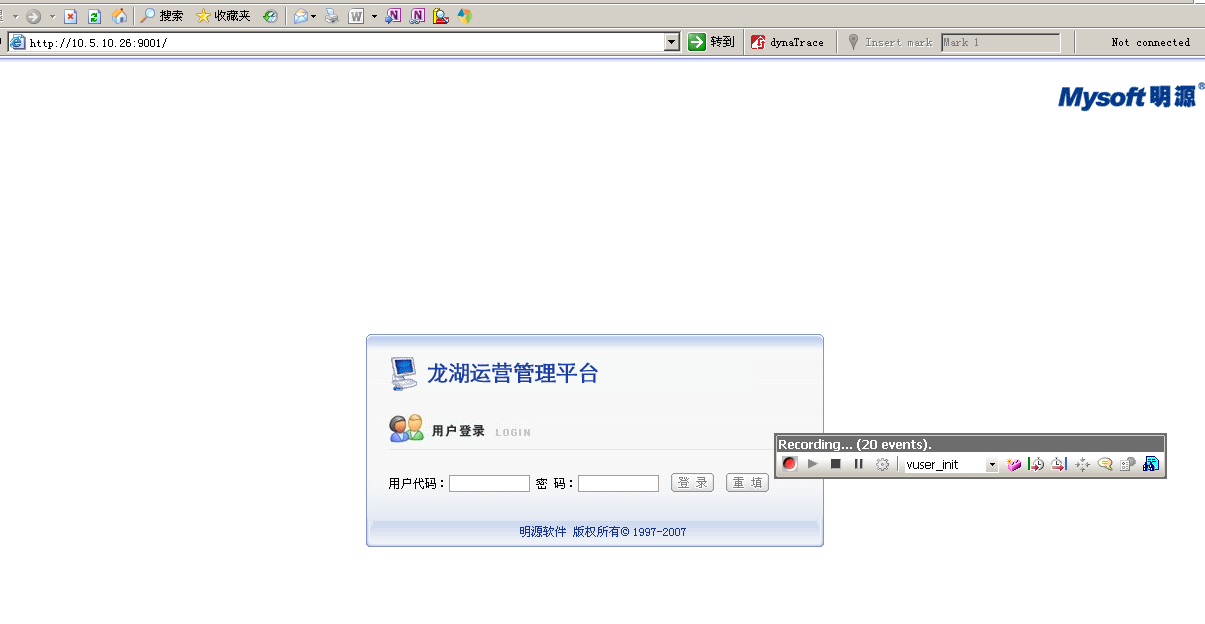


最后，我们来关注HTTP Properties Correlation，这里用来设置自动关联，我们取消勾选，后面会介绍如何进行手动关联

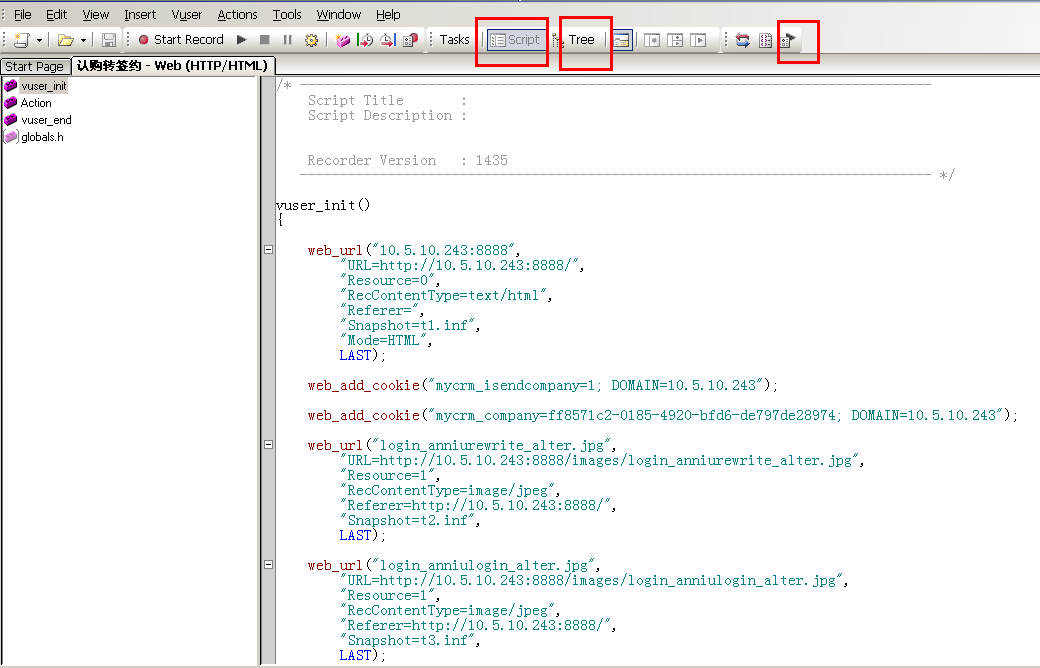


设置完成之后，我们就开始录制了。

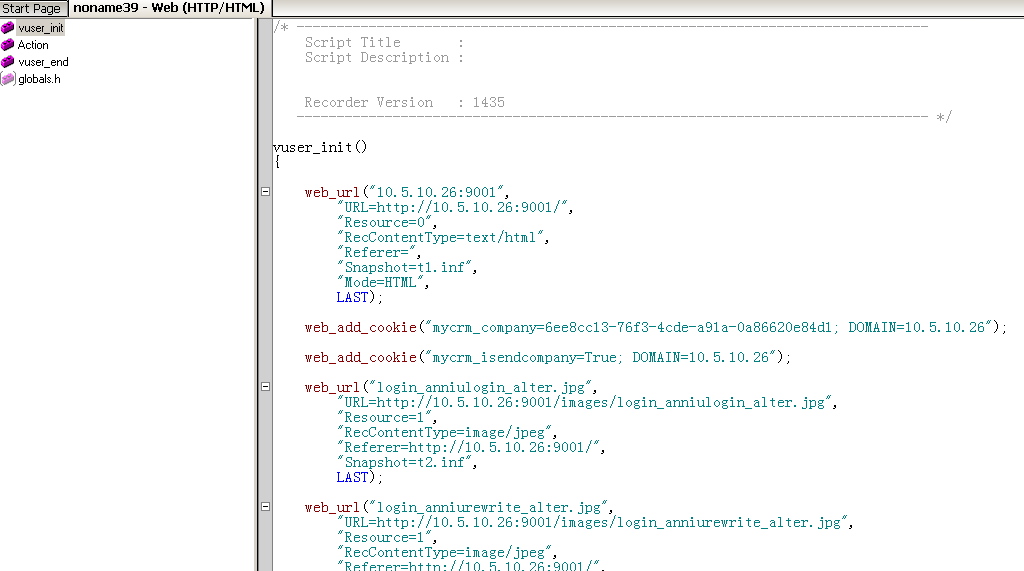
五、开始录制，如下图，录制的时候，我们可以擦入事务和集合点，集合点的图标在这里显示灰色，是因为在vuser\_int里面无法设置集合点，只能在Action里面设置（目前我们已不在使用集合点了，过强的并发会导致大量的响应过长，使系统性能的调优工作难度加大，也与实际的业务相差太远。业界虽然大部份公司出于各种考虑会用到集合点，但我们的目的比较单一，就是产出结果，让相关人员进行优化工作）。事务的擦入，我们按一个操作一个事务来进行（这个我们会在性能测试用例里面写到），一是为了方便进行脚本调式，二是为了更好的统计每步操作的性能。登录系统时我们用init，进入操作后，我们会一直用action直到退出。vuser\_end，我们很少用，比如有注销的地方，可以用一下，我们系统是直接关掉了IE，所以可不用。录制完成，关闭IE后，会根据录制的日志骤步生成脚本。



六、录制完成后，我们会看到如下的界面，界面中打了三个红框的就是我们常需要用的功能：Script、Tree、Run-time settings



首先我们来看Script和Tree中的主要内容



脚本中左边目录出现的globals.h，是头文件，这个我们可以不用管。

整个脚本中，我们可以看到两个主要的函数web\_url和web\_custom\_request

* Web\_url主要组成：
* URL：通过URL地址（资源统一定位器）要下载的web资源
* Referer ：发送下载请求的页面。比如登录ERP后的我的桌面，要下载图片，那么我的桌面就是referer
* Mode ：录制的等级，是否是HTML或者HTTP模式
* ENDITEM ：列表中每个资源的结束标志符
* EXTRARES后的资源是由script、active、java applet、flash、CSS产生的请求
* web\_custom\_request主要组成：
* URL：页面地址。
* Method ：页面的提交方式，POST或GET。
* TargetFrame：包含当前链接或资源的frame的名称。
* EncType：编码类型。
* RecContentType：响应头的内容类型。
* Referer：发送下载请求的页面。
* Body：请求体。

其它：

Web\_url根据函数中的URL属性加载对应的URL，不需要上下文。而web\_custom\_request需要上下文，比如我们的系统在web\_custom\_request经常会看到Session、oc2sale等。

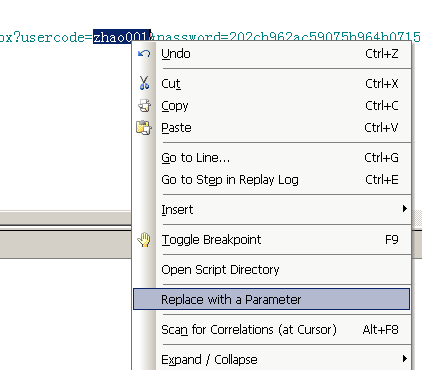
还有其它很多录制的函数我们就不介绍了，其实脚本主要需做好两件事

* + 参数化：一般情况下，我们需要输入的地方，就是要参数化的地方，但目前有些做关联的地方，我们也做了参数化，因为参数化脚本跑起来会更稳定些。所谓输入的地方，比如：登录的账号、登录密码、查找时输入的数据、修改时填入的数据等。所谓关联的地方我们也做了参数，比如：登录用户的CODEGUID、修改业务时的GUID等等，这些我们可以在数据库中直接查到的字段数据。具体关联的地方是用关联还是用参数化，和个人习惯还有调式脚本都有关，不强作规定。
  + 关 联：关联就是系统返回的数据，我们需要重新拿来请求的，在我们的ERP系统中，关联的绝大多数是各种GUID。

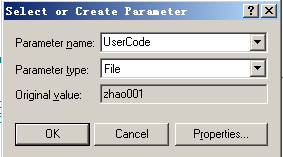
现在我们来进一步了解一下参数化的设置和关联的设置

#### 4.3.1.2 参数化

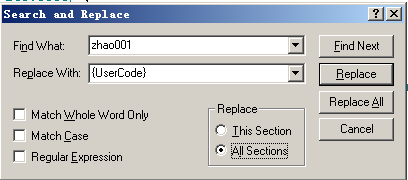
一、右键点击需要参数化的数据，然后选择Replace with a Parameter



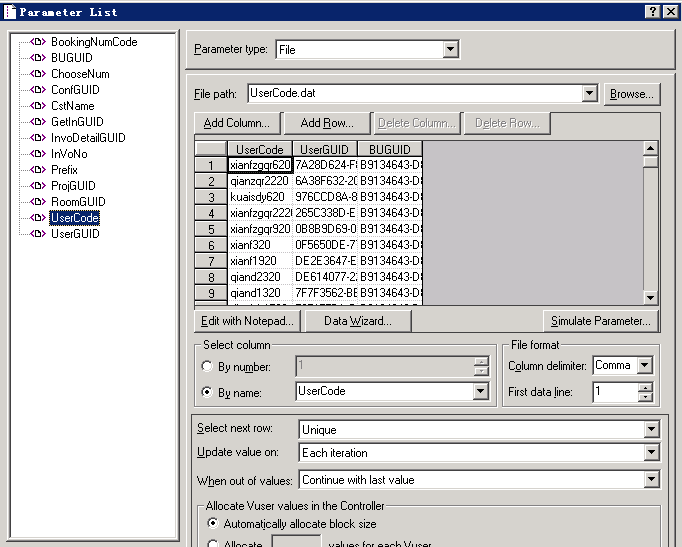
二、弹出下面的框后，将Parameter\_name的名称改为比较有意思的名称，最好是和数据库中对应的字段名称相同，这里我们关联的是usercode的值，所以我们这里就叫UserCode。然后，点击OK就行了。



三、OK后，脚本中的zhao001就变成了{UserCode}，并以粉色显示，我们需要替换掉脚本中的所有zhao001，我们用Ctrl+H，调出替换工具，Replace我们选All Sections，因为我们要替换所有脚本中的zhao001。准备好之后，我们点击Replace all即可。



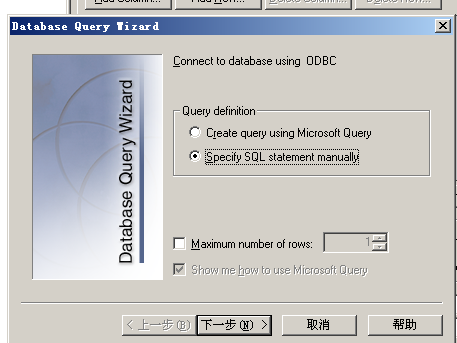
四、替换完成之后，我们点击工具栏中的Open Parameter List（快捷键Ctrl+L）



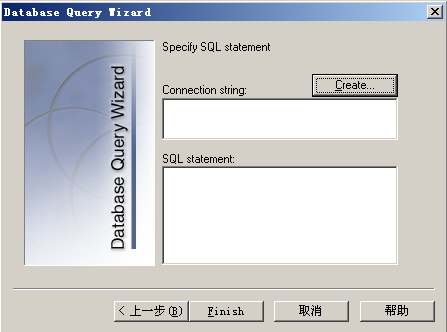
这里我们只介绍几个重要的地方

* File path：是参数文件名，多个参数可以共用同一个参数文件，比如参数UserCode、UserGUID、BUGUID我们用了UserCode.dat里面的不同列，这是因为他们在数据库中属于同一张表，或相关表
* Edit With Notepad：是用来进行手动编辑的，点击后可以直接将dat文件用记事本方式打开，我们做好数据后，可以在这里进行手动编辑，不过我们常用的还是Data Wizard
* Data Wizard：数据向导，可以直接从数据库中取数据，优点在于不用担心格式问题，只需要将找到的数据中的{}花括号替换成空就行了
* 数据向导操作如下：

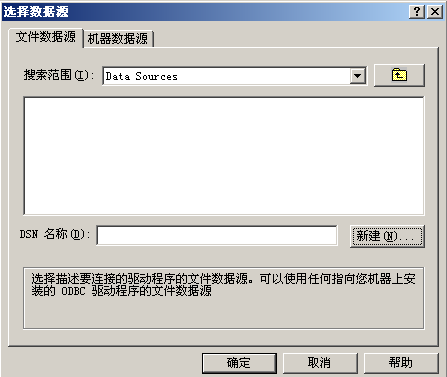
这里我们选Specify SQL statement manually选项



点击下一步，弹出如下框



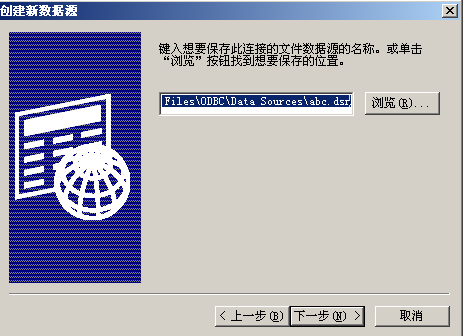
点击Create



点击新建



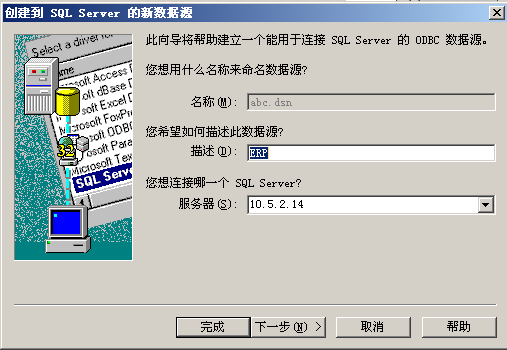
选择数据源为SQL Server，并点击下一步



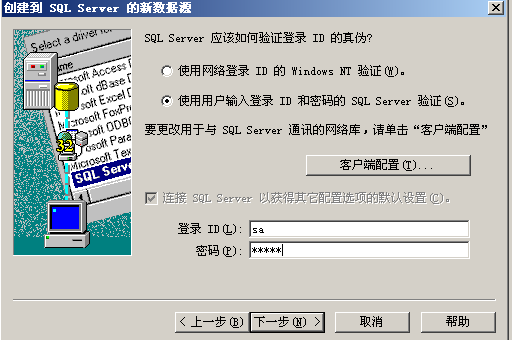
选择数据源创建的路径，然后下一步



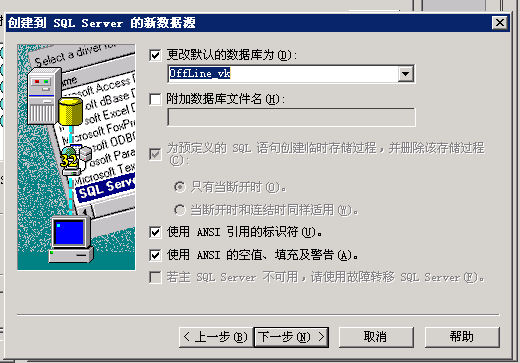
点击完成



输入描述和DB服务器IP和实例（没实例就只输IP），点击下一步



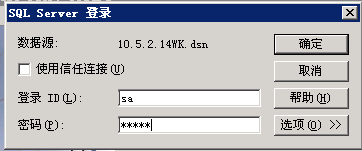
按图设置好登录方式，并输入SQL Server登录用户和密码，点击下一步



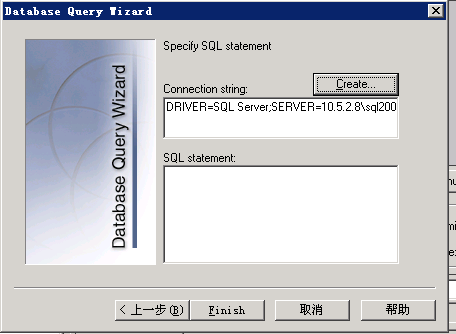
更改默认数据库后，点击下一步，然后点击完成。



点击一下测试数据源,看是否成功，成功后点击确定



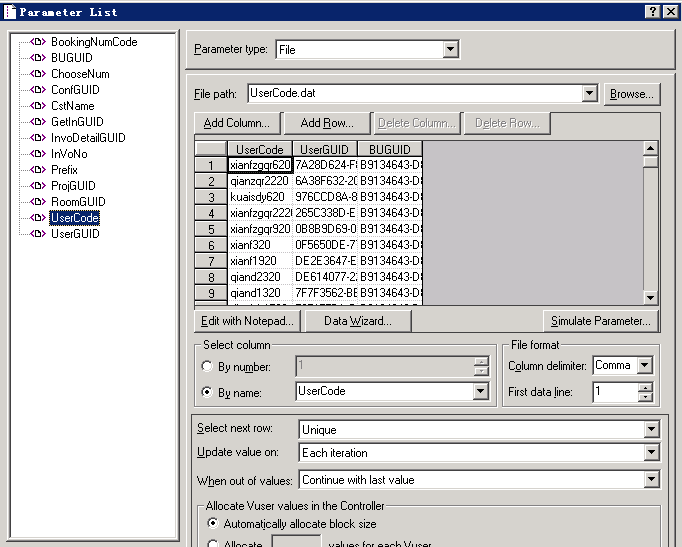
再次输入ID和密码，点击确定



将预先写好的参数化SQL，填入SQL statement里，点Finish即可

注意：1、SQL里面不能有GO，不然会报错。2、需要查找的数据行数不能过多，上次我查找4000条报错了，这种情况就只能手动在库里找到数据进行编辑了。

完成后，我们又回到这里了



五、Select column我们一般选择By name。

六、select next row、Update value on：这里的两个选项，分别有3个小选项，共6个小选项。

select next row下的选项分别是：Sequential(顺序)、Random(随机)、Unique(唯一)

Update value on 下的选项分别是：Each iteration(每次循环)、Each occurrence(每次出现)、Once(一次)

这里我们用比较好理解的方式解释一下这两大类选项。select next row针对的是取参数方式，Update value on针对的是参数改变的条件。从设置上理解前者重点在于怎么取，后者重点在于什么时候改变。从编程上理解，前者是针对对象，后者是针对方法，比如20个用户进入房间做事，那么这20个用户是怎么进入的，那就用Select next row设置，进入之后用什么方法做了些什么，就是Update value on设置。如果还是不太理理，我们就记三种常用的组合，下面我们来介绍这三种常用的组合。

* 默认Sequential+ Each iteration：怎么取？顺序的取。什么时候改变？每一次循环就改变。单场景并发我们可以用这种设置，因为在单场景中，我们没有环循，每个用户只需要顺序的取。
* 常用Unique+ Once：怎么取？唯一的取。什么时候改变，不改变。单场景并发我们也可以用它，组合场景中有了循环，那么登录的用户是不需在action循环的时候再去改变的，所以我们也可以用它。

(这种设置要保证参数够用，虽然不够用时，可以用When out of values进行设置，但我们推荐参数要够用)

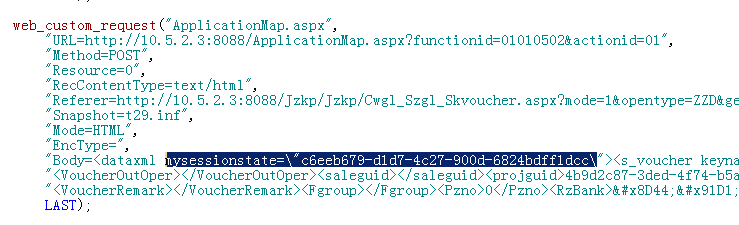
* 常用 Unique+ Each iteration：怎么取？唯一的取。什么时候改变，每次循环改变。组合场景我们可以用它，每个用户登录系统后，可以要做多笔业务，每做一笔业务我们需要重新循环一次做下一笔业务，那么我们需要改变业务数据，所以我们用这种设置

(这种设置要保证参数够用，虽然不够用时，可以用When out of values进行设置，但我们推荐参数要够用)

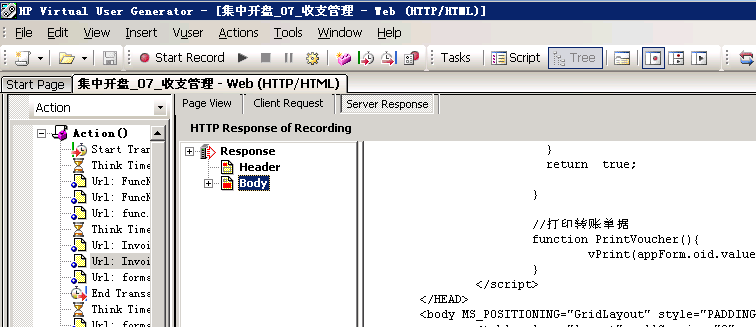
至于参数具体是怎么分配的，有兴趣的同学可以查找下资料，由于篇幅的限制，我们就不详细讲了

#### 4.3.1.3 关联

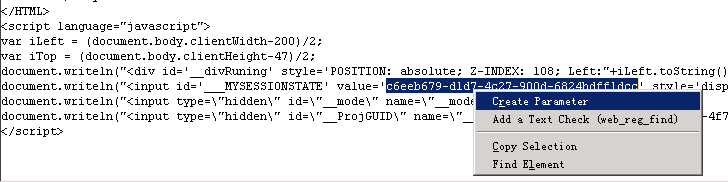
一、我们得先找到要关联的数据，比如Session，我们发现脚本的请求中会用到Session，Session是临时产生的，所以不可能提前通过SQL在表中找出作为参数，所以我们需要关联



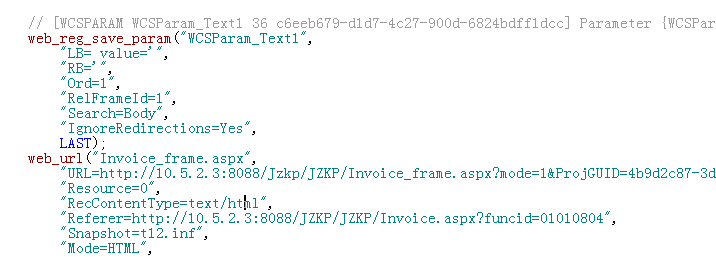
二、关联的时候，我们先切换到Tree,再在左边导航栏找到可能出现服务器返回Session的页面（这个用几次就能较快的定位了）,然后选择”Server Response”，并找到Body。因为有可能存在多个页面都返回了该值，但最好是在第一次出现返回的页面进行关联



三、只要选中需要关联的数据，右键选择“Create Parameter”，最后选择是即可



四、关联完成后，我们可以在刚才关联的URL页面前找到该关联函数



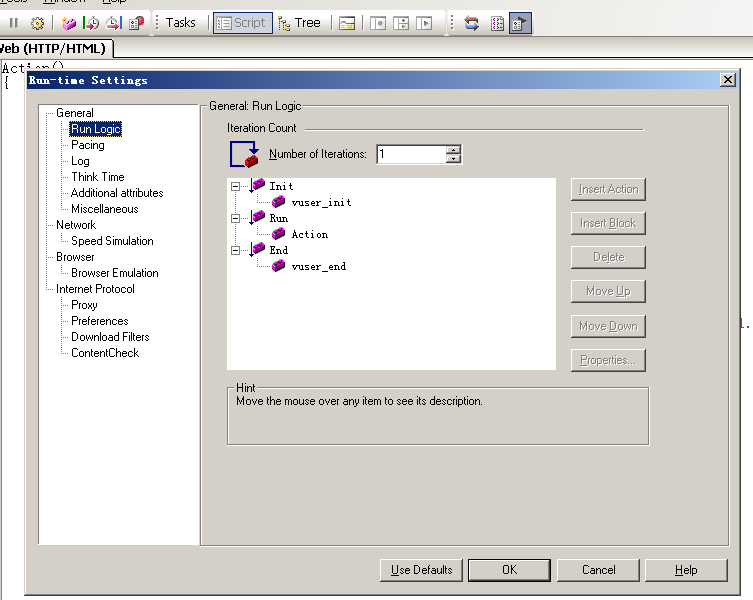
现在我们来介绍一下web\_reg\_save\_param函数

* WCSParam\_Text：是关联参数名，也是存放服务器返回值的地方。在需要用到该返回值的地方，我们将其替换成{WCSParam\_Text}，如4.3.1.3中第一步图里面的SESSIONGUID。关联参数我们也需要取有意义的名字。比如这里我们可以将WCSParam\_Text1改为SessionGUID
* LB：是参数左起字符。关联时会自动填上返回值左边的字符，如我们关联的SessionGUID的左边就是value=’。这里一般我们默认采用自动生成的，但有时候这里我们需要手动修改，一是当左边没有字符时，这里可能乱码，我们需要手动将乱码删除。二是当Ord后面的值较大时，一般超过5，我们就手动修改，将左边的字符增多，使左边唯一。
* RB：是参数右止字符。其它同LB
* Ord：关联参数所在的位置。关联函数通过LB和RB配对去查找关联内容，但有时会存在LB和RB在返回的页面中出现多组的情况，关联函数就会将这些值按顺存放在数组中，Ord就表示了我们需要的参数在数组中的位置。

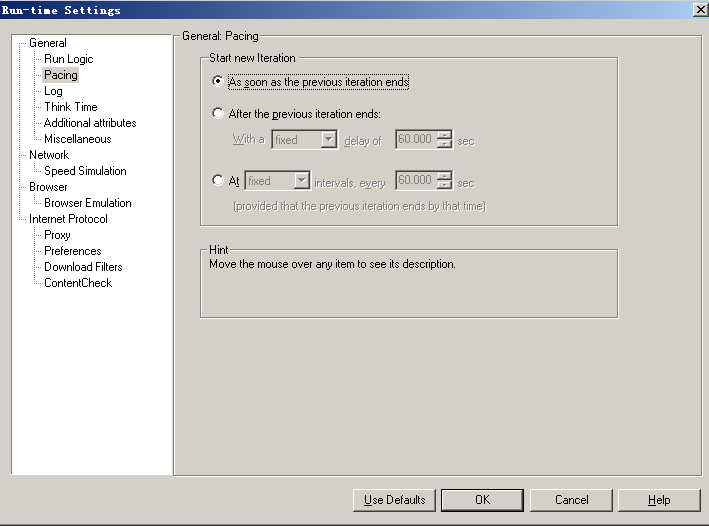
#### 4.3.1.4运行时设置

Run-time Settings，是脚本进行场景运行时的最要设置，我们经常进行修改的有Run Logic、Pacing、Log、Think Time 、Miscellaneous

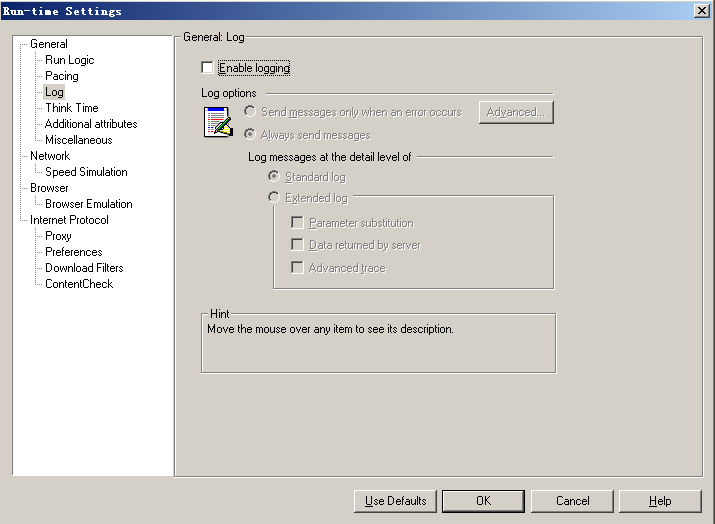
* Run Logic：主要是用来设置Action循环次数的，不能对Init和End进行循环次数设置。单场景并发我们这里设置为1，组合场景时，我们这里的值为根据每个用户的业务量来进行设置，业务量是多少，就循环多少次（这里的计算，我们会在实战中讲解）。



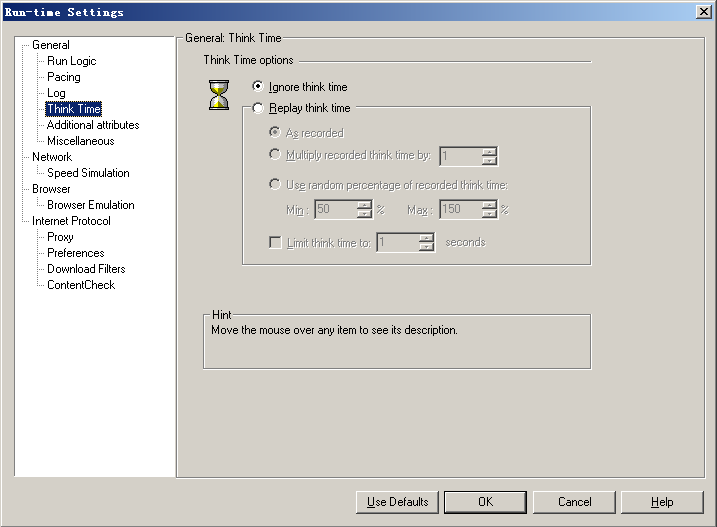
* Pacing：步进设置。有三种设置，第一种是前一次action运行完，马上运行下一个；第二种是前一次action运行完之后的XX秒，运行下一个；第三种是前一次action运行完开始，到下一个运行开始之间的时间。单场景并发我们选的是默认的第一种，组合场景时，我们选的是第三种，时间通过单位时间除以单位时间每个用户的业务数得出（这里的计算，我们会在实战中讲解）。



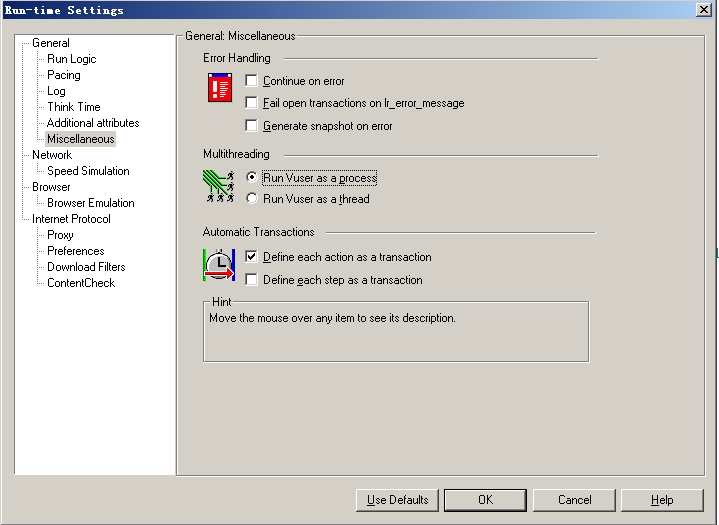
* Log：日志记录。我们一般取消勾选，不记录日志



* Think Time：思考时间。单场景我们选择第一项不忽视思考时间。组合场景我们选择第二项的第一个，即使用录制的思考时间



* Miscellaneous。这里我们只需关注Multithreading，进行线程和进程设置。我们在运行脚本的时候，会产生mmdrv.exe的进程。选择Run Vuser as a thread时，只会生成一个mmdrv.exe进程，每一个用户就是这个里程里面的一个线程。选择Run Vuser as a process时，会产生多个mmdrv.exe进程，每一个用户就是一个进程，每个进程占用内存8M左右。第二个选项更加稳定，所以我们在内存足够的情况下，推荐使用。IP欺骗也和这个有点关系，后面我们会讲到



#### 4.3.1.5 脚本调式

做完参数化和关联后，我们需要对脚本进行调试。我们通过回放进行调试，我们主要看脚本是否能回放成功和回放数据是否正确。脚本回放成功就是不报错，能正常回放。回放数据正确就是通过编写的检查语句查看新增、修改、删除的是否正确，当然数据是否正确，我们还要检查脚本回放时查询操作是否真正按要求进行了查询，我们可以通过SQL Server Profiler进行跟踪。

查看关联是否获取成功，我们用lr\_output\_message(lr\_eval\_string ("{NewCstGUID}"));函数进行调试。（lr\_eval\_string是获取用的，lr\_output\_message用来输出获取的内容）

脚本的调试是性能测试中的重要一环，这里我们也很难用较短的篇幅讲完，希望大家能在日常的工作中慢慢掌握规律

因为录制脚本的地址，有可能和我们最终测试的地址不一致，所以我们需要将访问的地址进行参数化，在脚本的最前面加上下面两个函数，并将脚本中的IP和端口分别替换为{ServerAddr}和{Port}

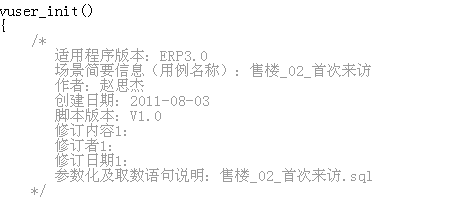
//服务器地址

lr\_save\_string("10.5.2.233", "ServerAddr" );

//服务器端口号

lr\_save\_string("8080" ,"Port");

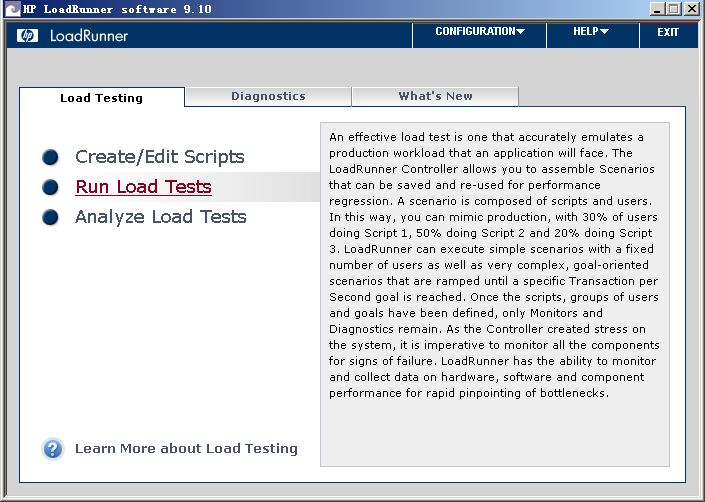
完成上面工作后，我们需要在脚本前面加上注释，以便他人查看



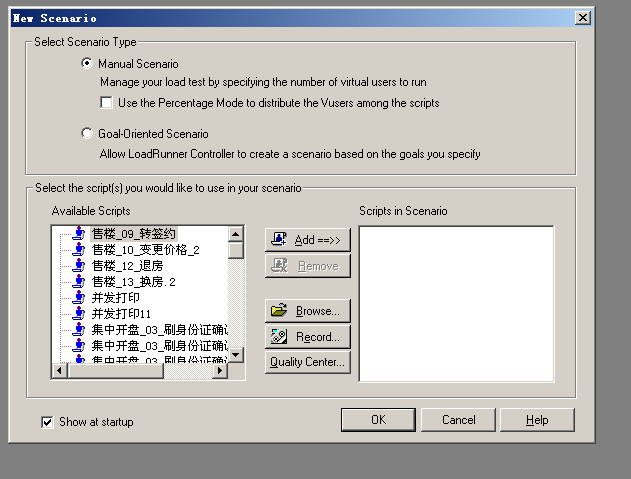
### 4.3.2 Controller使用

完成上面的脚本录制和调试后， 使用Controller将这个执行脚本的用户从单人转化为众人，从而模拟大量用户操作，进而形成负载。Controller可以帮助我们对场景的设计、执行及监控进行管理。

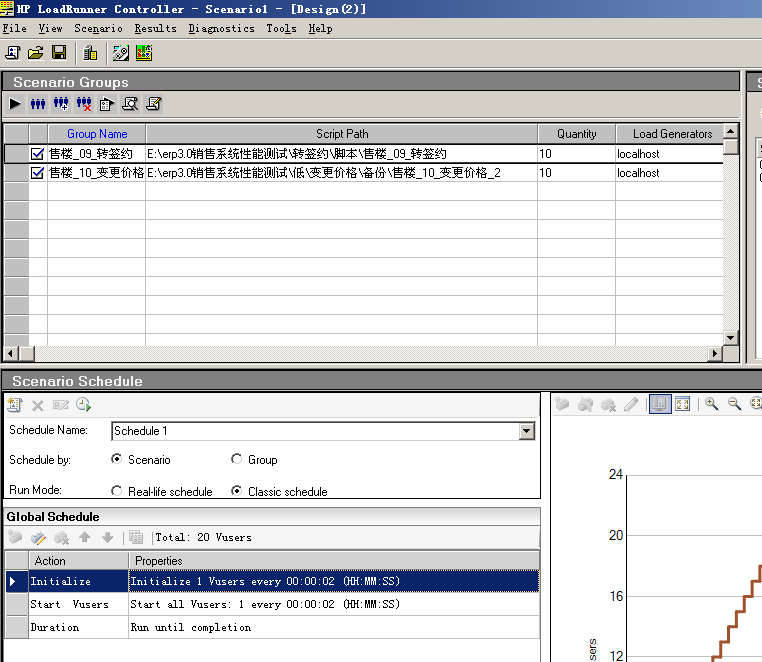
一、选择Run Load Tests



二，进入New Scenario，选择Manual Scenario（手工场景）。Add需要运行的脚本。点击OK



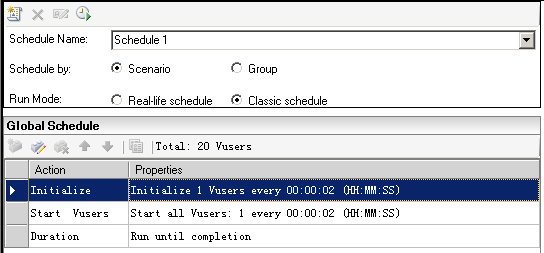
三，进入场景设置界面



场景设置是一个强大的功能集合体，可以模拟大多数的场景，这里我们只介绍我们工作中需要用到的设置，对其它设置有兴趣可查资料学习

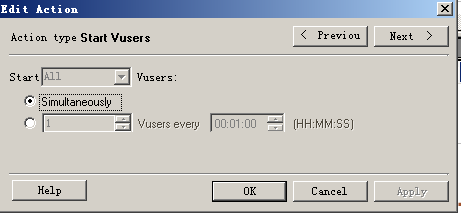
现在我们先来看Scenario Schedule(场景安排)。其中有Schedule by和Run Mode两种设置

Schedule by：列表设置。这里的列表指的就是脚本列表，有两个选项， Scenario和Group。Scenario，按场景设，指的是将脚本列表中的所有脚本当成一个整体进行设置。Group，按组设，指的是将脚本列表中的脚本进行单独设置。不管是单场景还是组合场景，我们现在的项目中都选的是Scenario。Scenario下对应的Run Mode有两个选项，分别是Real-life schedule和Classic Schedule。 前者是真实场景设置，选择之后可以很灵活的对脚本进行自定义设置，但我们一般选择Classic Schedule经典模式。经典模式最为简单，也足够我们现在项目上的使用。如下图



选择Scenario+Classic schedule之后，我们发现脚本的并发用户可以直接手动修改了。现在我们来介绍一下Initialize、Start Vusers、Duration

* Initialize：初始化，指脚本用户的初始化，双击打开后里面的设置会影响到Start Vusers的设置，一般情况下这里我们默认，不进行修改，我们直接改Start Vusers即可。
* Start Vusers：如下图有Start “all” Vusers，在经典模式下，这里不可改，表示所有的用户都参与下面的设置。设置中包含Simultaneously和 XX Vusers every XX。Simultaneously选项表示同时，即所有用户同时开始执行脚本。XX Vusers every XX，可自由设置用户的加载，如填写为 2 Vusers every 00:00:3,表示每3秒加载2个用户。

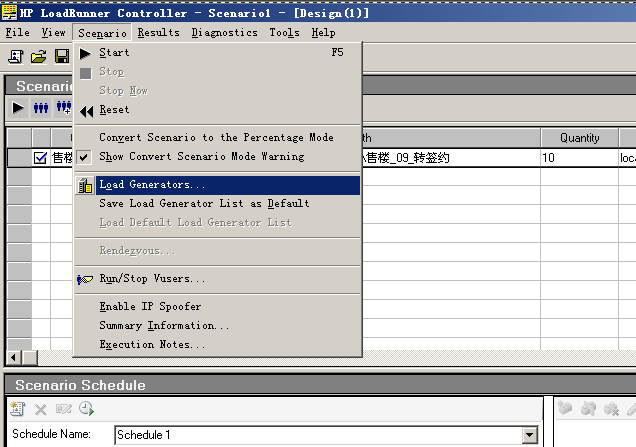


* Duration：运行时长设置。此功能选项，我们也默认不修改，改的过长会导致脚本参数不够用，运行时长我们会通过步进时间来控制。

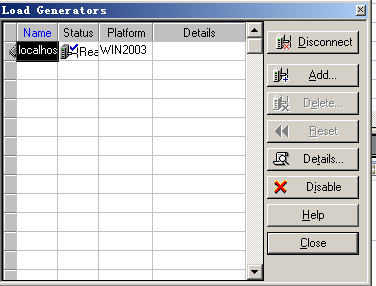
Controller中也有Run-time setting的设置，在VuGen中的设置会带到场景中，但在场景中修改的Run-time setting是不会影响脚本的设置的，只会对场景有影响

### 4.3.3 Load Generator使用

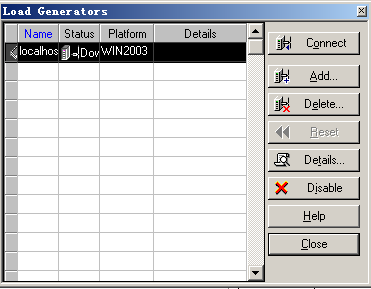
对场景进行设计后，接着需要对负载生成器进行管理和设置。Load Generator是运行脚本的负载引擎，在默认情况下使用本地的负载生成器来运行脚本，但是模拟用户行为也需要消耗一定的系统资源，所以在一台电脑上无法模拟大量的虚拟用户，这个时候可以通过多个Load Generator来完成大规模的性能负载



一、点击Load Generators..，打开如下界面



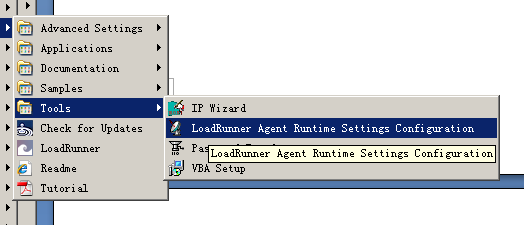
二、Disconnect点击后会变成Connect，同时连接会变成断开。选中相应的主机，点击Connect，就可以连接到相应的主机进行虚拟用户模拟



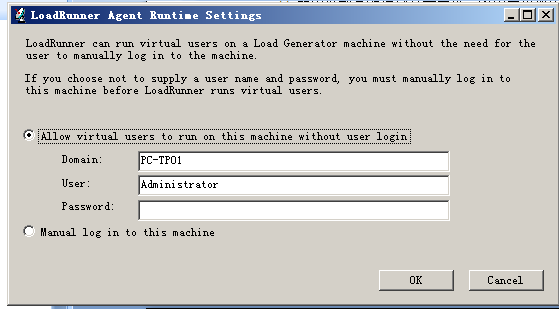
我们现在重点介绍一下Add…

Add..的用途就是链接到其它主机，以用来模拟用户，但使用Add..之前，必须链接的主机支持代理，也是就被链接主机安装了Load Runner，同时开启了代理，如何开启其它机器的代理呢？

首先，在目标代理主机的开始里面找到Load Runner的Tools，在Tools下有一个雷达一样的图标，LoadRunner Agent Runtime Settings Configuration：代理运行时设置

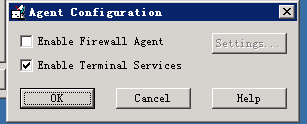


点击该设置，打开如下界面

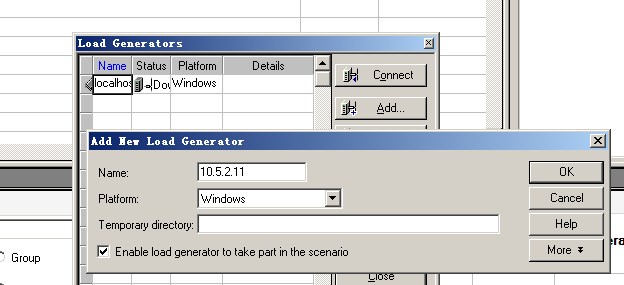


默认已经输入好主机名和管理员账号，只需要输入当前主机的密码，点击OK，最后会在电脑右下角出现雷达的图标，表示开启了代理。

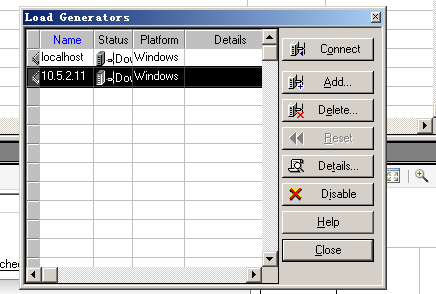
注意：默认选第一项后，在这里点击OK，可能开启不了小雷达，目前在我们实际的工作中已经出现了这样的问题，后来我们找到了解决的方法。开启不了小雷达，我们选择第二项，点击OK，发现会报运行时错误，报错后我们点击报错中的退出，然后进入开始-所有程序-LoadRunner-Advanced Settings,找到Agent Configuration，点击后弹出如下界面，我们勾选第二项后，点击OK，发现小雷达出现了。



三、出现小雷达后，回到我们的主负载机，也就是回到第二步的最开始，点击Add，出现如图所示

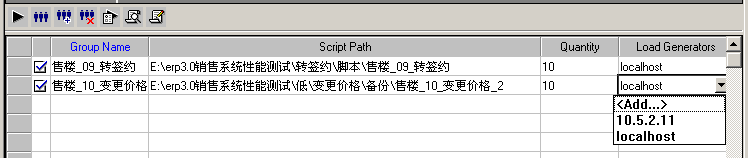


四、输入目标代理主机的IP地址，点击OK，弹出如下图所示

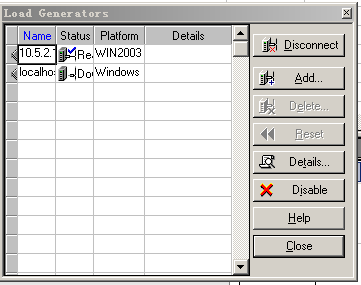


五、选中目标主机，点击Connect，可以发现目标主机被链接上，目标主机中的雷达边也多了个云

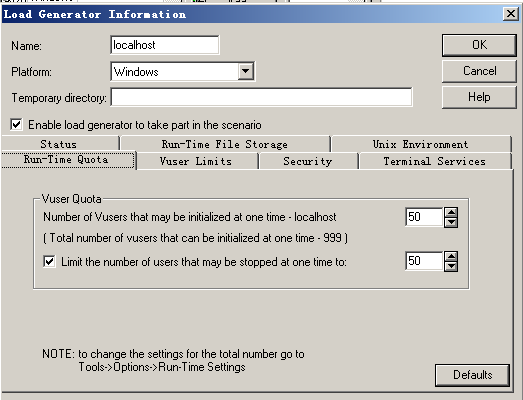
六、连接上了，我们就可以对各主机进行虚拟用户分配了，只需要在Load Generators的下拉列表中选中对应的主机即可



Load Generator中还有一个很重要的设置。就是用来设置init人数的，我们在运行脚本的时候会发现，在场景监控中，init默认不会超过50个人，也就是最大并发是50个人，我们想使用超过50个人的并发，就需要在这里进行设置了

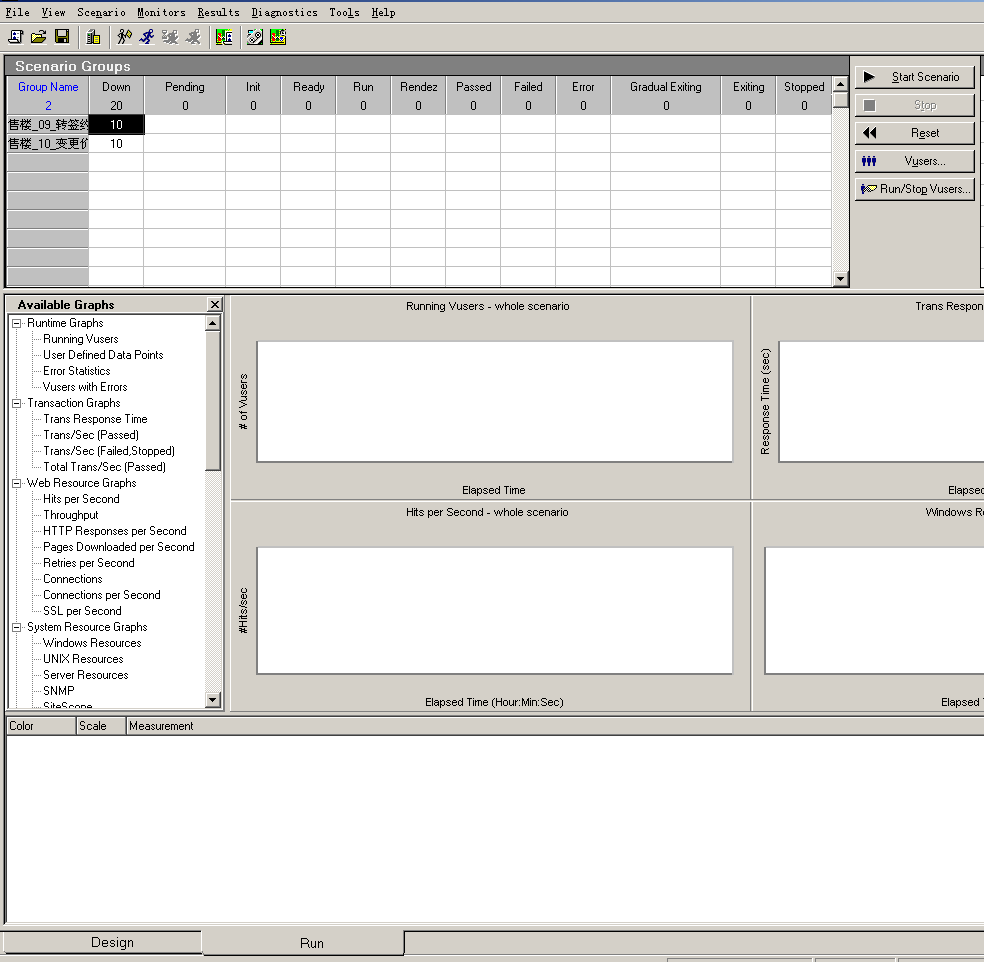


点击Details，弹出如下图，找到Run-Time Quota标签页。两个值都默认为50，第一个值init最少加载多少用户开始运行脚本，第二个值指加载多少用户就停止加载，开始运行脚本，不勾选表示初始化所有用户后开始运行。不过这里的两个值，我们一般只在单并发时才进行修改，比如单并发100个用户，那么我们把两个值都设置为100。组合场景目前的加载方式我们是每秒加载两个，所以不需要修改这里



### 4.3.4 场景运行

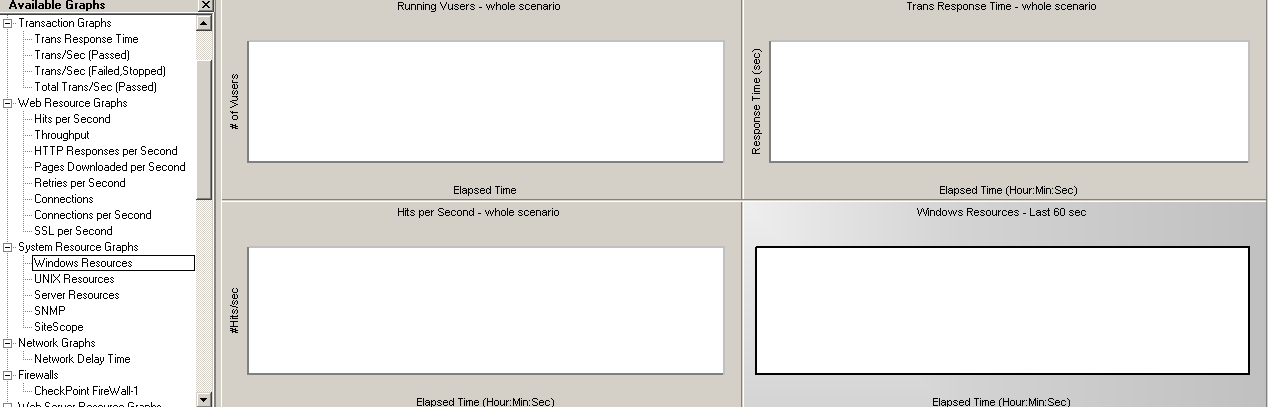
完成上述工作后，我们就可以进行场景运行了，切换到Run，点击Start Scenario，场景就开始运行了。



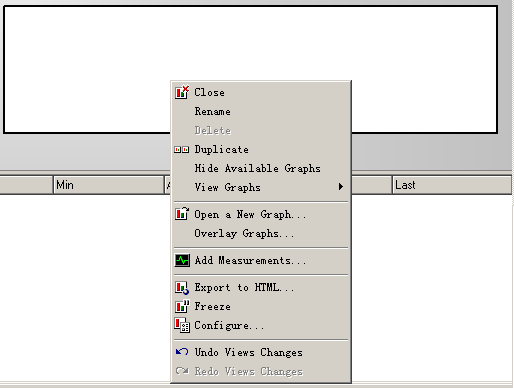
Controller 能够监控系统资源并不是因为安装了LoadRunner的缘故。在没有安装LoadRunner前，它也可以得到当前系统的相关性能数据。比如任务管理器中的CPU和内存使用率、各个进程所使用的相关资源信息等，这些都是系统在设计中留下的指标接口。几乎所有的应用和服务软件都提供了计数器和监控器接口，在SQL Server中可以通过事件探查器完成SQL命令执行的监控工作。

我们来增加一个系统资源组的Windows Resources计数器：

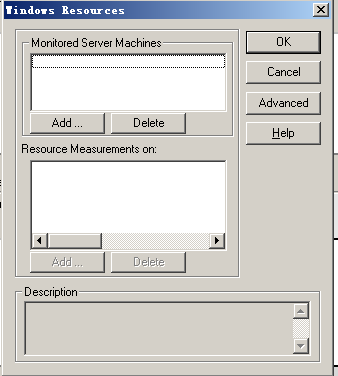
首先我们双击左边系统资源组的Windows Resources，右边监控窗口会出现Windows Resources窗口



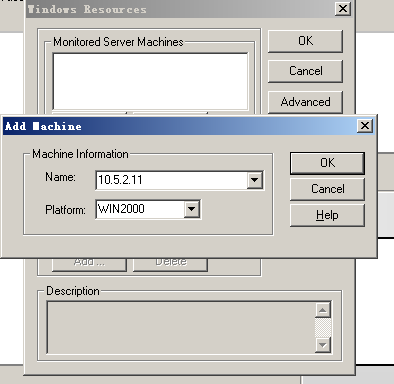
然后我们右建点击该窗口，选择Add Measurements



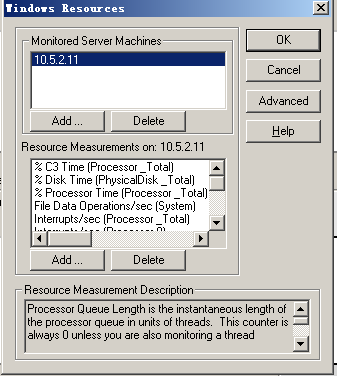
弹出如下窗口



我们选择Add



输入目标主机IP地址，点击OK，回到如下窗口，我们可以通过最下面的Add去添加计数器，我们推荐使用默认的计数器，直接点击OK就可

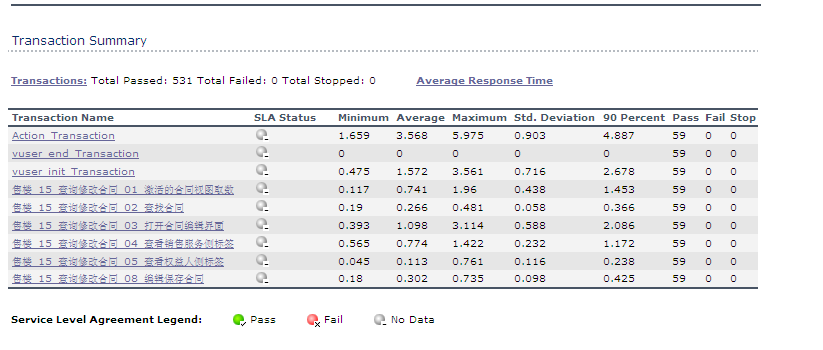


场景中的计数器的功能非常强大，有兴趣可翻阅资料学习，虽然现在我们公司测试人员对这一块不承担分析工作，也没有要求在这里添加计数据，但不排除今后项目过程中会用到。

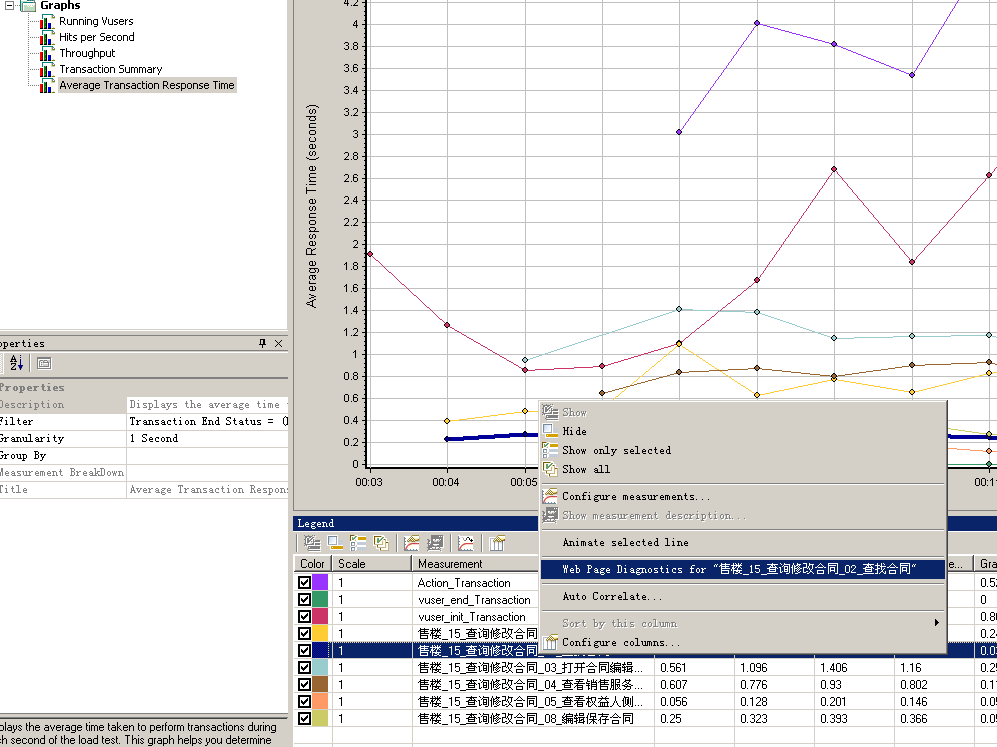
### 4.3.5 Analysis使用

通过场景完成负载后，我们完成了性能测试的执行过程，接着就是通过负载的结果来发现和定位性能瓶颈。目前我们公司定位性能瓶颈主要由性能优化人员来做，所以这里我们就不做详细介绍了，相关知识可以查资料学习，基本的一些东西我们还是需要掌握的。本段只说我们在测试报告中需要收集的数据，当然如果相关人员有需求，我们也要增加数据收集的。

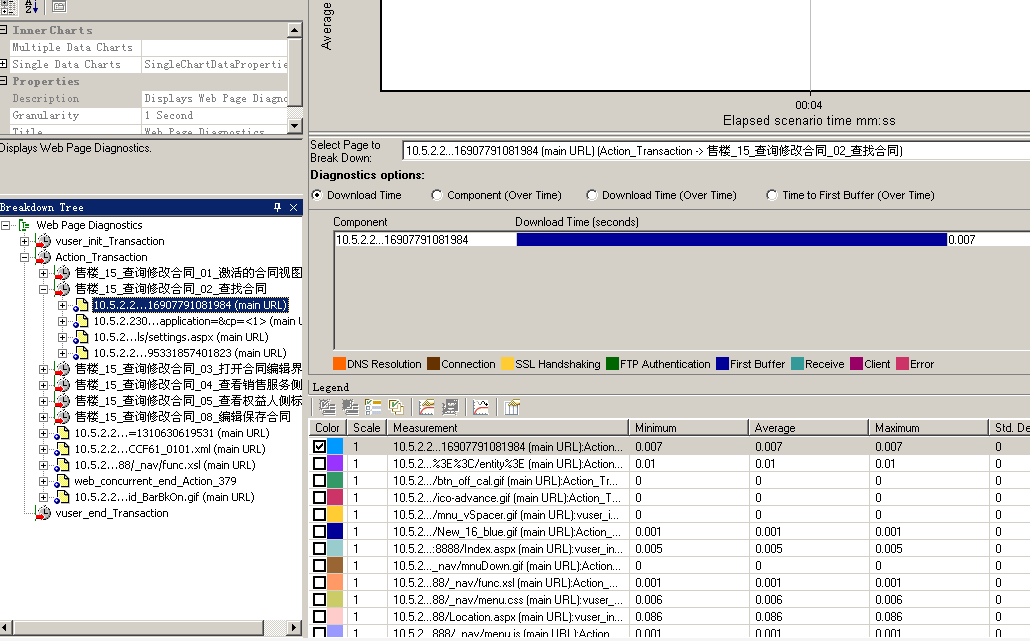
首先我们要收集的就是Summary Report概要报告，我们需要收集其中事务对应的最小值、平均值、最大值、90%值、通过数



其次，如开发有要求需要具体到超时的URL的情况时，我们可以通过以下步骤来查找。找到Graphs下的Average Transaction Response Time平均事务响应时间，点击后找到对应的事务右键



选择Web Page Diagnostics页面诊断，进入如下界面，双击事件中的URL，就可以看到该URL对应的页面诊断了。通过这个图，可以对事务的组成进行抽丝剥茧的分析，得到组成这个页面的每一个请求时间分析，进一步了解响应时间中有关网络和服务器的处理时间的分配关系。



诊断选项：

* Download Time 下载时间分析。这里可以得到组成页面的每个请求下载时间。
* Component(Over time) 各模块的时间变化。这里列出组成页面的每个元素，以及随着时间的变化所带来的响应时间的变化。
* Download Time(Over time) 模块下载时间变化。这里提供了针对每个组成页面元素的时间组成部分分析，方便确认该元素的处理时间组成部分。
* Time to First Buffer(Over time) 第一次请求时间。这里会列出该元素所使用的时间分配比例，是受Network Time影响的多还是Server Time影响的多。

（First Buffer：建立连接后，从Web Server 发出第一个数据包，经过网络传输到客户端，浏览器成功接受到第一字节的时间就是First Buffer。这个度量时间不仅可以表示Web Server 的延迟时间，还可以表示出网络的反应时间）

页面响应时间组成：

* Client Time：客户端浏览器接收所需要使用的时间，可以不用考虑。
* Connections Time：连接服务器所需要的时间。
* DNS Resolution Time：通过DNS服务器解析域名所需要的时间，解析受到DNS服务器的影响。
* Error Time：服务器返回错误响应时间。
* FTP Authentication Time：FTP认证时间，这是进行FTP登录等操作所需要消耗的认证时间。
* Receive Time：接受数据的时间，这个时间反映了带宽的大小，带宽越大，下载的时间越短。
* SSL Handshaking Time：SSL加密握手时间

（SSL握手协议（SSL Handshake Protocol）：它建立在SSL记录协议之上，用于在实际的数据传输开始前，通讯双方进行身份认证、协商加密算法、交换加密密钥等）

其它分析器我们就不介绍了，有兴趣的查资料学习

### 4.3.6 IP欺骗

如果我们部署了负载均衡，我们就需要使用IP欺骗，启用该技术后，场景中运行的虚拟用户将模拟从不同的IP 地址发送请求

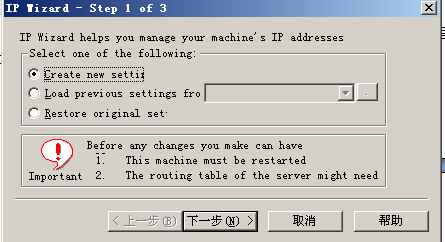
使用IP Spoofer 的步骤:

1. 使用IP Wizard：在“开始”菜单程序中，找到LoadRunner🡪Tools🡪IP Wizard，运行IP Wizard

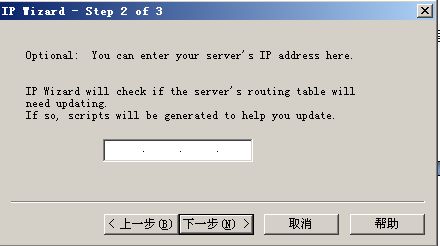
注意：运行IP Wizard 程序的机器必须使用固定的IP，不能使用动态IP。

IP Wizard选项说明：

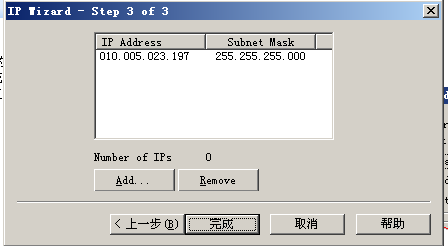
* Create new settings：第一次运行IP Wizard 需要选择该项来增加新的IP。
* Load previous settings from file：选择保存好的文件，如果以前运行过IP Wizard，可以选择该项。
* Restore original settings: 此项用于使用IP欺骗进行测试完成后，释放IP 的过程（因为该机会占用大量的IP资源，可能会导致其他机器没有IP可用的尴尬局面，使用该项，可以恢复到原来的状况）。



点下一步，出现IP Wizard-Step2 of3的窗口



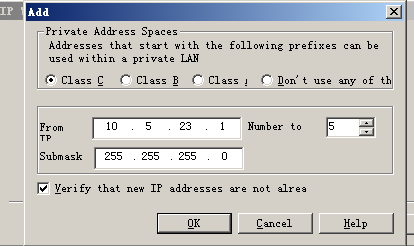
直接点下一步，出现向导的第三个窗口 IP Wizard-Step3 of3



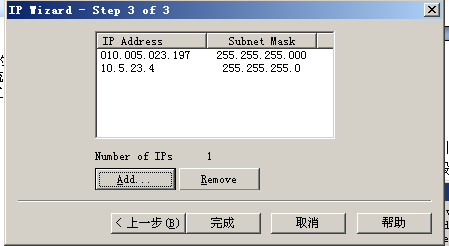
点击Add进入添加IP的窗口，在“From IP”文本框中输入要使用IP 范围的第一个IP 值，然后在“Numbers to Add”输入一 个数字，表示IP 范围的值；假如第一个IP 为 10.5.23.1，范围大小为5，那么IP Wizard 将会使用10.5.23.N（1<=N<5），当然这个范围内已经使用的IP地址除外

，否则会引起IP 冲突。“Submask”采用默认情况即可，取决于使用的那种类型的网

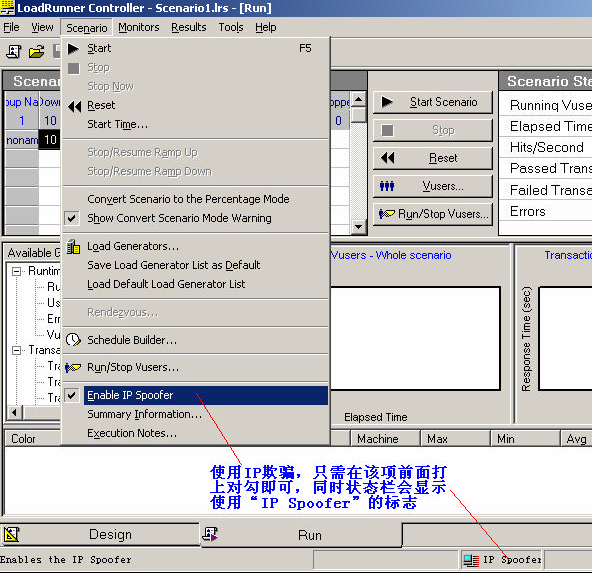
络IP，一般局域网内采用Class C 即可。



然后“OK”，然后IP Wizard 开始检查该范围内没有使用的IP，并把没有使用的IP添加到本机的IP 窗口中。到最后一个窗口，直接点“完成”，使用IP Wizard 后，最后重新启动机器。

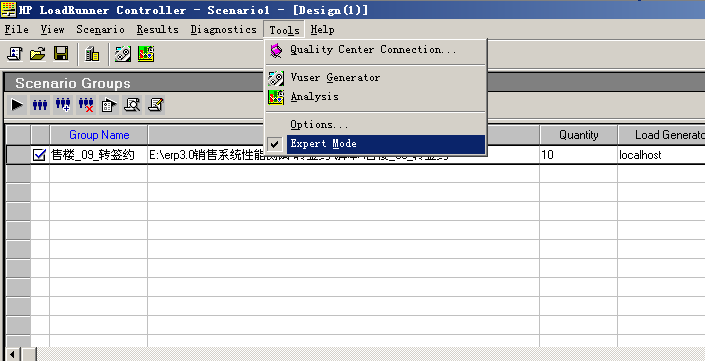


在Controller 的场景 中，将菜单：Scenario->Enable IP Spoofe打勾即可启用IP Spoofer,启用后，Controlle r的状态栏里会显示IP Spoofer标志

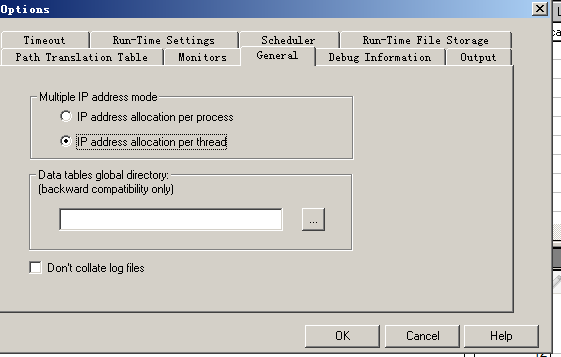


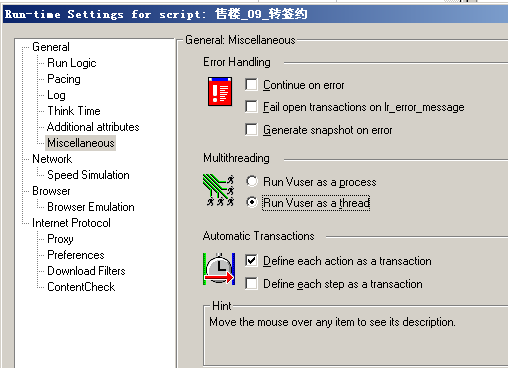
注：完成以上步骤，我们还需要查看是否勾选专家选项，以及专家选项中的IP欺骗的线程进程设置是否与脚本一致，如果不一致，可能使IP欺骗失效，操作如下：

勾选Tools下面的Expert Mode专家模式



再点击Tools下的Options，打开如下界面，找到General标签页，将IP地址分配方式设置与脚本中的run-time settings一致





# 5 性能相关知识学习

## 5.1 交换机限速

万科电子开盘项目，我们采用了交换机限制网速进行大数据量测试。通过产品经理调研，客户现场的光纤为8Mbit，有20台电脑在上网使用，于是可以算出平均每台机器的网速为0.4Mbit，即将交换机网速限制为512Kbit。

要限制交换机端口网速，首先我们要知道该交换机有没有IP，普通交换机是不支持设置IP的，但好一点的是可以支持的，这个可以问公司的IT人员，并问清交换机的IP地址和需要限速机器对应的端口。

交换机限速如下：

第一步：telnet 交换机IP，链接到交换机，输入账号、密码。账号密码输入时不显示任何信息，也不支持退格删除。

第二步：输入system-view，这个命令我在网上查是IBM的SystemView网络管理系统，是企业级的

第三步：输入interface 端口号。这里就是要进入目标机器链接的交换机端口了

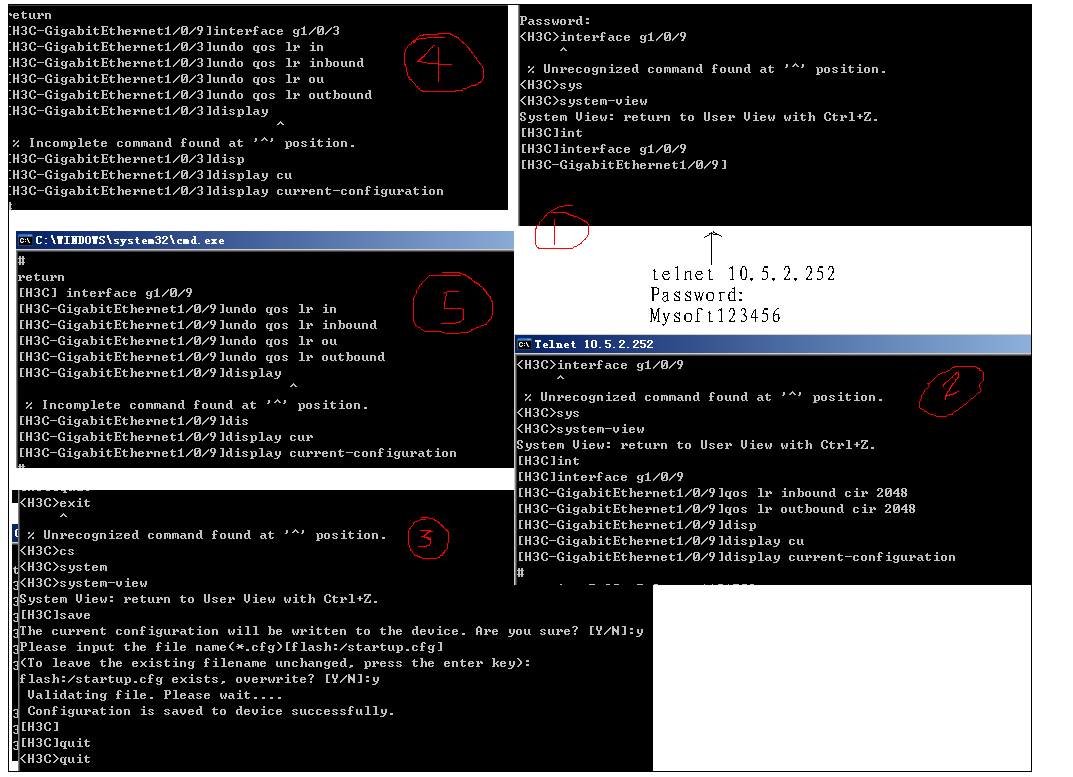
第四步：输入qos lr inbound cir 网速、qos lr outbound cir 网速。前者命令是限制目标机器上传的，后者命令是限制目标机器下载的

第五步： 输入display current-configuration，查看设置是否成功

第六步：退出quit

设置完成，我们用机目标机器后，需要将其限速改回来，可以用命令undo qos lr inbound来放开。

在命令行中我们可以输入一部分命令后，按”Tab”键，系统会自动为我们输入未完成的部分。比如我们输入syste，然后按”Tab”键,系统会自动完成到system-view



## 5.2 WEB服务器与负载均衡

ERP3.0后，我们的产品就完全支持负载均衡了，支持负载均衡，开发层面要有文件同步和缓存同步，所以我们需要向项目经理确认是否需要做负载均衡。部署指引请看“负载均衡部署指引”



## 5.3 DB快照与SQL索引优化

快照设置方法如下：



SQL索引优化这一块较强大，了解了对我们的测试也有帮助，我在网上收集了较好的学习资料，供参考学习



## 5.4 数据缓存与计划缓存

数据缓存：数据缓存是数据库数据在内存中的临时容器，它包含了数据库表数据在内存中的拷贝，位于数据库与数据访问层之间。删除、更新、增加数据的时候，同时更新缓存。

计划缓存：在执行一个存储过程或查询之前，SQL Server需要先进行编译和优化，选择一个执行计划，编译之后的计划会被存储在计划缓存中。

简单的描述两者区别：数据缓存就是存放表数据的缓存，计划缓存就是存放被编译好了的SQL语句的缓存。

一般测试情况下，我们只清除数据缓存。

--清除缓冲区缓存（数据缓存）

--DBCC DROPCLEANBUFFERS

--清除执行计划缓存

--DBCC FREEPROCCACHE