

HP LoadRunner 软件 — 配置、脚本编写和执行的提示与技巧

白皮书



目录

摘要	3
作者简介	3
执行成功的性能测试计划	3
制定高级测试计划	4
脚本编写提示	5
定义目录结构和命名规则	5
记录脚本	5
通过关联性和参数化创建动态脚本	5
创建事务	5
用参数表示并转换 itoa (C++) 代码	6
Date_Time 操作	6
获得当前日期七天后的日期	6
轮询批处理运行状态	7
记录消息	7
为检查表编写脚本	7
构建情境提示	8
通过 HP LoadRunner 控制器执行测试	8
时间安排选项	8
数据处理提示	9
使用虚拟表	9
针对具有相同子网掩码 IP 段和网络的客户端的 IP 欺骗	9
解析基于数值的字符串	10
用 switch 和 case 语句进行随机化	10
测试执行和时间安排提示	11
HP LoadRunner 超时错误	11
扩展后的日志代码	11

摘要：本指南提供 HP LoadRunner 软件的配置、脚本编写和执行的相关提示与技巧。它是 HP LoadRunner 超级用户 Opral Wisham 所学课程的综合，其中包括特有代码以及从其他测试人员处收集的代码。本指南旨在帮助测试人员学会使用 HP LoadRunner，并为长期使用 HP LoadRunner 的人员提供新的最佳实践。

作者简介：Opral Wisham 目前是 Waste Management, Inc.（行业领先的综合废料管理服务提供商）在性能和容量方面的专家。

Waste Management 的总部位于德克萨斯州的休斯顿，它确保了许多社区的生活质量，为北美洲的大约 2500 万居民及 200 万商业客户提供废料解决方案。

Opral 在 Waste Management 的职责包括管理公司所有新应用的容量和性能验证 (CPV) 流程。在担任 CPV 技术经理期间，Opral 收集了很多最佳实践，用于协调公司跨部门 CPV 团队的测试活动，并加快对团队新成员的培训。

执行成功的性能测试计划

在将任何新的应用投入生产之前，必须执行涵盖范围广泛的容量和性能验证 (CPV) 测试。本指南旨在帮助新用户和经验丰富的专业人士学习利用 HP LoadRunner 制定并成功实施负载测试计划的新方法。

性能测试流程可以分为以下四个阶段：最初的性能测试请求、测试准备、脚本开发和执行，以及测试分析。

本文档的以下章节描述了部分步骤以及在制定并执行有效的性能测试计划时可以遵照的最佳实践。

表 1. 该表阐述了性能测试执行流程的功能流程，并附有各个阶段的建议交付成果。

	CPV 请求	准备	开发和执行	分析
描述	<ul style="list-style-type: none"> 经过 CPV 后，将对总体解决方案进行测试 召开计划会议，概述需求并解决问题 保留与 CPV 之间的测试间隙 	<ul style="list-style-type: none"> 安全硬件 配置硬件 / 软件 数据爆炸 创建 ID 业务情境和测试用例 	<ul style="list-style-type: none"> 创建自动化脚本 调试自动化脚本 针对每个测试周期运行预先确定的负载 在每个测试周期后提供结果 提供总结报告 	<ul style="list-style-type: none"> 分析 CPV 为每个测试周期提供的数据 确定不可接受的性能 确保硬件容量准确
所有者	业务所有者	业务所有者 管理员	CPV	业务所有者 CPV
可交付成果	<ul style="list-style-type: none"> 架构审核委员会批准 性能需求文档 (草案) 已确定的支持人员 	<ul style="list-style-type: none"> 性能需求文档 (最终版) 配有支持人员 已配置的测试环境和已验证的数据 获取自动控制工具 (如未提供) 	<ul style="list-style-type: none"> 总结报告概括了调整变更、CPV 问题和统计数据 	立场声明和推荐

制定高级测试计划

性能测试团队应首先定义高级测试计划，以描述所有测试工作的时间安排，其中包括要执行的测试类型（例如，在线的基础架构压力和负载测试、批处理性能测试等）。此外，该计划还应指明性能测试团队将如何与企业内部的开发、部署和/或支持小组配合展开工作。在可能的情况下，该计划有助于在说明交互作用的过程中使用资源的实际名称和标题。

测试团队应描述测试意欲测量和/或报告的对象。阐明所有可能造成误解的术语（例如，“事务的响应时间”（“Transaction’s Response Time”）可以指从按下“enter”按钮到从系统得到结果所需的时间，或是指执行某项需要人为干预的功能所需的时间）很重要。某些用于 Web 应用的量度与传统的在线批处理应用具有不同的定义，如平均点击数/秒、Web 服务器吞吐量等。

下列检查表可在定义综合性能测试流程的步骤时使用：

1. 列出项目各个阶段（试验、代码冻结和生产阶段）的所有测试里程碑和可交付成果。
2. 创建综合生产的物理结构图，使每个细节的信息都一目了然，包括每个组件的连接方式。
3. 列出测试所需的所有硬件和软件需求。
4. 确定所有数据库的数据量需求。
5. 定义所有性能目标。
6. 确定所有性能基准（如果有）。
7. 确定预期用户总数。
8. 指定并发用户数量的最大值和最小值。
9. 选择五个业务情境（人工编写的脚本），在每个应测量的情境（事务）中加入动作。
10. 确定用户负载混合。
11. 定义每个事务可接受的平均响应时间。
12. 定义平均点击数/秒或一小时内的事务数。
13. 定义吞吐量（如果适用）。
14. 确定所有正在运行的并发批处理流程。
15. 确定所有领导、技术和联系人员。

以下章节提供的示例可帮助性能测试团队简化脚本编写、情境建立、数据处理、测试执行及测试时间安排的流程。

脚本编写提示

定义目录结构和命名规则

为创建目录结构和制定命名规则确定正式的程序至关重要。为新项目创建目录结构（文件夹名称）时可以使用以下命名法：

```
ProjectName
├── LR_Scripts
├── Scenarios
├── TestDoc
│   ├── TestPlanAndReport
│   ├── TestExecution
│   └── Recommendations
├── Reports
│   ├── LR_Reports
│   │   ├── Cycle1
│   │   └── Cycle2
│   └── HTML_Reports
│       ├── Cycle1
│       └── Cycle2
```

对于系统的新用户而言，项目和脚本的名称应直观且具有描述性。例如：

```
ProjectName: 应确定进行测试的项目或应用。
LR_Scripts: 文件应为已保存的 HP LoadRunner 脚本的名称。
FunctionName_Type
    类型：为 func、rpt 或 qry
Scenarios: 文件应确定模拟的项目和虚拟用户的数量。
ProjectName_VuserCount
TestDoc: 目录应包含所有测试文档。
TestPlanAndReport
    测试计划
    测试用例
    工作量等
    脚本注释
    测试总结
TestExecution
    执行进度日志
    对比日志
推荐
    推荐
```

记录脚本

性能测试团队应尽可能地使用 HP LoadRunner 的自动记录选项。该方法提供了最高效的途径来记录大多数基于 Web 的应用的负载测试脚本。

通过关联性和参数化创建动态脚本

性能测试团队可以在前述语句需要某个值之前利用关联性来检索该值，从而使服务器能动态生成未知值。当值已知且在执行语句之前不可用时，性能测试团队可以进行参数化处理。

下面是参数化的示例：

1. 利用文件将“来源”、“货币”、“业务单元”和“帐户”参数化。选择的下一行应为“与业务单元相同”。“业务单元选择”的下一行应为“连续”或“随机”（这取决于持续时间或重复次数）。
2. 利用 %m/%d/%Y 将日期用参数表示为“date/time”。
3. 将“数量”用参数表示为 1 到 10,000 之间的随机数。
4. 利用 %m 将“记帐期”用参数表示为“date/time”。
5. 用文件将 URL 参数化。这样可以减少 URL 变更时所需的脚本变更。

关联性和参数化是创建动态数据驱动测试的关键。

创建事务

然后，性能测试团队应定义用于测量服务器性能的事务。每个事务都会测量服务器对指定虚拟用户 (Vuser) 的请求做出响应的的时间。

为了标记事务的开始点，测试团队应在记录工具栏中单击“启动事务”按钮。在事务名称框中输入事务名称。单击“确定”接受事务名称。HP LoadRunner 虚拟用户生成器 (VuGen) 会自动将 lr_start_transaction 语句输入脚本。单击“结束事务”按钮来结束事务。VuGen 会自动将 lr_start_transaction 语句输入脚本。

通常，性能测试会为登录、注销及脚本中出现“submit”的任何地方创建事务。可以包含更多事务，但所有提交都必须具有事务以测量响应时间。

动态事务名称是指根据参数化值的变化而变化的名称。如果存在多个随机运行的报告，团队可以使用下面的代码动态地定义事务名称：

```
Char rname[10];
sprintf( rname,
        "Run_Report_%s",
        lr_eval_string("{report}"));
lr_output_message(rname);
lr_start_transaction(rname);
lr_end_transaction(rname,2);
```

用参数表示并转换 itoa (C++) 代码

如果团队需要增加计数器数值或将计数器作为参数使用，则需要利用 sprintf 函数将整数转换为字符串。转换后，需要将字符串用作参数。因此，应该用 lr_save_string 将字符串保存为参数。

```
/* M:\Remedy\Test Scripts\Modify an Existing Remedy Help
Desk Case */
/* insert before action */
char wid_str[6]; /* 0xFFFF = 65535 + NULL = 6 */
int i = 0; /* initialize i */

i = i + 2; /* increment i by 2 or whatever value */
sprintf( wid_str, "%d", i); /* copy integer into string */
lr_save_string(wid_str, "wid_num"); /* save into parameter value */
```

Date_Time 操作

可以插入以下语句来操作日期时间戳。在本例中，脚本中的日期必须为当前日期后一天的日期。参数 {Tomorrow} 保存在脚本中出现当前日期的地方。插入该参数后，会在出现当前日期的地方插入明天的日期。

```
Action1()
{
    lr_save_datetime("Tomorrow is %m/%d/%Y", DATE_NOW +
ONE_DAY, "Tomorrow ");
    //lr_output_message(lr_eval_string("{Tomorrow }"));

    itemdata.....
    *
    *
    *
    "Name=PV_REQ_SR_WRK_DUE_DT$0", "Value={Tomorrow }",
    ENDITEM,
```

获得当前日期七天后的日期

此代码会在当前日期的基础上加七天。如果交付日期在当前日期的七天之后，这会很有用：

```
lr_save_datetime("%m/%d/%Y", DATE_NOW + ONE_DAY +
ONE_DAY + ONE_DAY + ONE_DAY + ONE_DAY
+ ONE_DAY, "curr_date");
```

轮询批处理运行状态

本例用于要求单击“刷新”按钮直到运行状态变为“成功”的脚本。此代码提供自动刷新，直到批处理作业完成。下一步要求完成批处理作业。

```
int x; // flag will be 0 or 9
char *temp, *temp2; // values to hold strings
Action1()
{
temp2="Success"; //compare string 2
//lr_message("temp2 = %s", temp2);
.....
// set x to 0 x is the success flag
x=0;
do {
web_reg_save_param("RunStatus",
"LB=<td align='LEFT'
class='PSLEVEL1 GRIDODDROW'
>\n",
"RB=\n",
"Ord=5",
"Search=body",
LAST);

web_submit_data("PROCESSMONITOR.PROCESS
MONITOR.GBL",
"Action=http://crpu028a:8050/psc/fs84cpv/EMPLOYEE/ERP/c/
PROCESSMONITOR.PROCESSMONITOR.GBL",
"Method=POST",
"RecContentType=text/html",

"Referer=http://crpu028a:8050/psc/fs84cpv/EMPLOYEE/ERP/c/
PROCESSMONITOR.PROCESSMONITOR.GBL?
Page=PMN_PRCSLIST&Action=U&",
"Snapshot=t17.inf",
"Mode=NORESOURCE",
ITEMDATA,
"Name=ICType", "Value=Panel", ENDITEM,
"Name=ICElementNum", "Value=0", ENDITEM,
"Name=ICStateNum", "Value={ICStateNum6}",
ENDITEM,
"Name=ICAction", "Value=REFRESH_BTN", ENDITEM,
"Name=ICXPos", "Value=0", ENDITEM,
"Name=ICYPos", "Value=0", ENDITEM,
"Name=ICFocus", "Value=", ENDITEM,
"Name=ICChanged", "Value=1", ENDITEM,
"Name=ICFind", "Value=", ENDITEM,
```

```
"Name=PMN_FILTER_WS_OPRID", "Value=CPVID",
ENDITEM,
"Name=PMN_FILTER_PRCSTYPE", "Value=", ENDITEM,
"Name=PMN_FILTER_FILTERVALUE", "Value=1",
ENDITEM,
"Name=PMN_FILTER_FILTERUNIT", "Value=1", ENDITEM,
"Name=PMN_FILTER_SERVERNAME", "Value=PSUNX",
ENDITEM,
"Name=PMN_FILTER_PRCNAME", "Value=", ENDITEM,
"Name=PMN_DERIVED_PRCINSTANCE",
"Value={Process_Instance}", ENDITEM,
"Name=PMN_DERIVED_TO_PRCINSTANCE",
"Value={Process_Instance}", ENDITEM,
"Name=PMN_FILTER_RUNSTATUS", "Value=",
ENDITEM, LAST);

// Compare correlation value with character string
temp = lr_eval_string(lr_eval_string("{RunStatus}"));
// correlation value to variable

//lr_message("temp = %s", temp);
//compare string 1

if(strcmp(temp,temp2)==0){
x=9; // set flag to indicate success
}
}while (x == 0); // do while flag not set
```

记录消息

使用消息功能之一将虚拟用户消息发送到虚拟用户日志和输出窗口。CPV 最常用的是 `lr_output_message`。日志消息可以用来确认脚本执行期间关于虚拟用户的许多事项，包括：

1. 确定虚拟用户在脚本执行期间是否到达特定点。
2. 确认向参数传递了有效的值。
3. 确认所传递的值是否出现在数据库的适当位置（方法是捕获所传递的值并了解要搜索的对象）。

为检查表编写脚本

在创建了所有自动化脚本后，HP LoadRunner 会进行脚本测试。下列情况下，可将脚本添加到性能测试台中：

1. 没有发生脚本编写错误。
2. 循环成功；用不同的值代替参数化变量。
3. 在每个系统层上记录活动。
4. 模拟至少有十个用户负载的情况。
5. 脚本在随后一天运行（日期关联正确）。

构建情境提示

通过 HP LoadRunner 控制器执行测试

可以使用以下步骤执行性能测试：

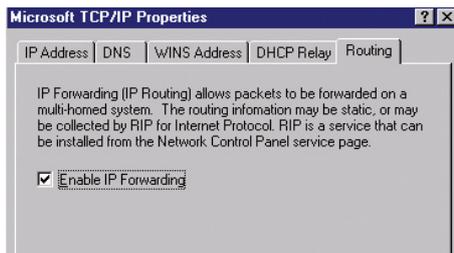
1. 添加负载生成器并连接。
2. 添加情境组：将一个或多个脚本添加到情境列表中。
3. 创建运行时设置：根据业务分析人员或用户的反馈来协调用户。
 - a. 禁用日志记录
 - b. 通过业务分析人员或用户来确定在脚本的哪些地方插入判断时间。如果脚本中有非随机值，则可以将判断时间随机化。
4. 模拟浏览器高速缓存：该选项指导虚拟用户用高速缓存模拟浏览器。高速缓存用于保存频繁访问的文档的本地副本，从而减少连接到网络所需的时间。默认情况下，高速缓存模拟处于启用状态。如果禁用了该选项，所有虚拟用户都不能用高速缓存来模拟浏览器。**注意：**与常规的浏览器高速缓存不同，分配给虚拟用户的高速缓存只会模拟图形文件的存储。该高速缓存不存储与网页相关的文本或其他页面内容。每个虚拟用户都有自己的高速缓存 - 每个虚拟用户必须从高速缓存保存和检索图像。禁用高速缓存后，HP LoadRunner 对每个页面图像仍只能下载一次。

5. 每次重复时，模拟一个新用户：指导 VuGen 在重复操作期间将所有 HTTP 环境重置为初始部分结尾处的状态。在该设置下，虚拟用户可以更准确地模拟新用户以启动浏览会话。它将重置所有的 cookie，关闭所有活动的连接，清空高速缓存，并重置用户名和密码（默认情况下，处于启用状态）。
6. 按情境安排时间：随着用户数量的增加，可能需要调整等待周期 (ramp-up)。大多数测试（压力测试除外）的持续时间应为一小时。
7. 按组安排时间：允许测试人员按组交错情境，这意味着情境 B 可以在情境 A 之后 10 分钟开始。等待周期是指每个情境中多个虚拟用户登录到系统的频率。持续时间是指特定情境组在等待周期后还有多久才会被执行。测试人员可以控制持续时间，使测试在相似的时间停止。

时间安排选项

有时，有必要将测试设置为稍后运行。该设置也可以在 HP LoadRunner 控制器中完成。要将测试设置为稍后运行，首先应按照其运行的原有方式设置测试。例如，在设计屏幕上调整运行时的设置（就如同利用未经安排的测试来完成一样）。然后，从菜单栏里选择 *情境/开始时间*。输入测试延迟时间（例如，现在是下午 5:00 点，测试应该在下午 7:00 点进行，则应输入两个小时的延迟时间）。

图 1. 在“网上邻居”（负载生成器上）中设置。



数据处理提示

使用虚拟表

脚本偶尔可以用来生成运行其他脚本所需的数据。此时，就必须使用虚拟表来完成该任务（此类任务要求对原始脚本做出最小程度的更改）。HP 虚拟表可提供将信息从一个脚本发送至虚拟表的功能，以便从其他脚本的虚拟表中检索信息。

示例：解压 C:\Program Files 下的文件包，并将 *vtclient.dll*、*vtserver.dll* 和 *vtst2.sh* 复制到相应的文件夹中。使用 *regsvr32* 并确保存在下列文件：

```
Regsvr32 "C:\WINNT\system32\msflxgrd.ocx"
```

```
Regsvr32 "C:\WINNT\system32\vtstcls.ocx"
```

确保“C:\WINNT\system32\COMCTL32.DLL”存在。

在控制器和将用作虚拟表服务器的机器上安装虚拟表。将 *vtclient.dll* 和 *vtserver.dll* 复制到 *lr/bin* 文件夹，然后找到“include”文件夹 (*lr/include*)，并插入 *vtst2.h*（这些都可以在硬盘驱动器上的产品文件夹中找到）。

针对具有相同子网掩码 IP 段和网络的客户端的 IP 欺骗

当需要唯一的用户 IP 地址来识别唯一的数据时，会用到 IP 欺骗。

在脚本中：

```
用于验证 IP 欺骗的代码：lr_message (" IP ADDRESS _ %s ",  
lr_get_vuser_ip());
```

添加 IP 地址：

转到 IP 向导……依照说明添加 IP

1. 开始菜单 -> LoadRunner -> 工具 -> IP 向导
 2. 检查创建新设置
 3. 输入 Web 服务器 IP (即 10.208.4.171)
 4. 从分配的 IP 地址 (即 10.208.40.4) 输入要添加的 IP 数目 (即 250)，然后检查验证未使用的 IP
 5. 单击“完成”。将出现包含 bat 文件 (nt 和 unix) 路径 (如 C:\temp\unix_routing.sh 和 C:\temp\nt_routing.bat) 的弹出窗口
 6. 检查重新启动
验证添加的 IP 地址
1. 开始菜单 -> 运行 -> CMD -> 输入 ipconfig。将显示所有可用的 IP 地址

控制器中的设置：

- 运行时设置 — 作为“进程”（常规选项卡中）运行
- 在控制器工具栏的“情境”选项卡中启用“IP 欺骗”
- 在控制器工具栏的“工具”选项卡中，选中“专家模式”
在“网上邻居”（负载生成器上）中设置：见图 1。

此外，验证所有新的 IP 地址是否都已在“网上邻居”/“属性”/“协议”/“高级”中列出。

验证 Wins 地址和 DNS 是否都已设置了正确的 IP 地址。

注意：

为了使 IP 欺骗起作用，IP SPOOFING ADD ROUTE BAT 文件需要在目标服务器上运行。

解析基于数值的字符串

利用下面的代码解析字符串并在随后的步骤中使用这些数据。字符串 "Wisham333445555" 可被划分为多个变量，如姓和规格序列号。

```
char *string;
char *first_x;
// LASTNAME=Wisham333445555
string = lr_eval_string("{LASTNAME}"); //Save value in
LASTNAME
//      lr_output_message("LastName: %s", string);
first_x = (char *)strchr(string, '3'); // Find first occurrence of
3 and save values after
// lr_output_message("SSN: %s", first_x);
      lr_save_var( first_x, 9, 0, "SSN"); // Save 9
characters after 3 encountered
// lr_output_message("SSN: %s", lr_eval_string("{SSN}"));
```

用 switch 和 case 语句进行随机化

用 switch 命令改变事务。如果正在测试多个报告，这些报告的 Web URL 有着明显区别但是其他步骤都相同，那么用户就可以使用 switch 语句。

```
int num;
num = ((rand () % 4 ));

switch (num)
{
case 0:
lr_think_time(300);
lr_start_transaction("FS_QRY_06_ZAM_ACQ_VERIFY_SEARCH");
:
:
break;
case 1:
lr_think_time(300);
lr_start_transaction("FS_QRY_06_ZLT_UDP_VERIFY_SEARCH");
:
:
break;
default:
lr_output_message("Random Number selected was not between
0 and 1");
}
```

测试执行和时间安排提示

HP LoadRunner 超时错误

如果希望实现某项功能（如保存数据）所需的时间少于 **1,000** 秒，则可以利用以下步骤来避免 HP LoadRunner 出现超时错误：

1. 从“控制器”转到“运行时设置”->“浏览器模拟”选项卡->“高级”
2. 将默认的超时设置（应为 3）更改为小于或等于 1000 的数字
3. 单击“确定”
4. 转至“工具”->“选项”->“超时”选项卡
5. 将负载生成器和 VU 超时参数更改为小于或等于 1000 的数字
6. 单击“确定”

如果希望实现某项功能（如上载）所需的时间大于 **1,000** 秒，则可以利用以下步骤来避免出现 HP LoadRunner 超时错误：

1. 修改 LoadRunner\template\qtweb\default.cfg。插入以下内容：

```
[web]
ReceiveTimeout=10000
```

2. 修改 LoadRunner\data\WebAdvancedOpt.txt。更改为：

```
[LIST]
1= "T_COMBOBOX;Web;EnableIPCache;DNS caching;将主机的 IP 地址保存到高速缓存中，以节省随后对该主机进行呼叫所需的时间。;1;Yes;<Yes;No>"
2= "T_COMBOBOX;Web;HTTPVer;HTTP version;指明应用所采用的 HTTP 版本。Select version 1.0 or 1.1.;1;1.1;<1.0;1.1>"
3= "T_COMBOBOX;Web;KeepAlive;Keep-Alive HTTP connections;允许持续的 HTTP 连接，以通过同一 TCP 连接发送多个请求。;1;Yes;<Yes;No>"
4= "T_EDIT_NUM;Web;MaxConnections;Concurrent connections limit;设置在脚本执行期间虚拟用户可以维护的并发活动连接的最大数目;1;4;<0;10>"
```

```
5= "T_EDIT_NUM;Web;ConnectTimeout;Connect timeout (seconds);出错前虚拟用户等待连接到服务器所花费的最长时间（以秒为单位）。;1;120;<0;10000>"
```

```
6= "T_EDIT_NUM;Web;ReceiveTimeout;Receive timeout (seconds);出错前虚拟用户等待从服务器接收消息所花费的最长时间（以秒为单位）。;1;120;<0;10000>"
```

```
7= "T_EDIT_NUM;Web;NetBufSize;Network buffer size;设置虚拟用户可以使用的网络缓冲区的最大容量。默认值 12288 个字节。;1;12288;<0;100000>"
```

```
8= "T_COMBOBOX;Web;CacheAlwaysCheckForNewerPages;高速缓存模拟总是会检查新的页面;;1;Yes;<Yes;No>"
```

```
9= "T_EDIT_NUM;Web;PageDownloadTimeout;页面下载超时（以秒为单位）;;1;120;<0;32000>"
```

3. 利用以上的步骤 1-6 来更改控制器 GUI 参数。

扩展后的日志代码

该代码打开跟踪，以扩展模式回放脚本的指定部分，而不以该模式回放整个脚本。这样就大大缩短了回放时间。

```
// TURN FULL TRACE ON
lr_set_debug_message(LR_MSG_CLASS_EXTENDED_LOG |
LR_MSG_CLASS_FULL_TRACE |
LR_MSG_CLASS_RESULT_DATA | LR_MSG_CLASS_PARAMETERS,
LR_MSG_ON);
```

```
web_reg_save_param("EMPLID",
"LB/IC='PSEDITBOX_DISPONLY' >",
"RB/IC=<",
"Ord=4",
"Search=body",
LAST);
```

```
// TURN FULL TRACE OF
lr_set_debug_message(LR_MSG_CLASS_EXTENDED_LOG |
LR_MSG_CLASS_FULL_TRACE |
LR_MSG_CLASS_RESULT_DATA | LR_MSG_CLASS_PARAMETERS,
LR_MSG_OFF);
```