

# Smartbi 工具集

- listsessions.jsp（会话信息）
- listpools.jsp（对象池信息）
- getclassurl.jsp（查找加载类路径）
- listwsdl.jsp（WSDL地址）
- threaddump.jsp（线程堆栈）
- jsconsole.js（JS控制台）
- heapdump.jsp（JVM堆栈）
- CPUMonitor.jsp（CPU时间）
- Memory.jsp（内存）
- listExtensions.jsp（加载扩展包列表）
- listobjects.jsp（系统对象个数列表）
- testmulticast.jsp（分组广播测试）

调试工具集，主要用于辅助开发人员定位问题。操作入口：[系统运维](#) > [调试工具集](#)。

## listsessions.jsp（会话信息）

该页面以XML的方式显示当前服务器上所有的会话信息，并列出各个会话中已经保存的会话属性（sessionAttribute）。可以通过该页面列出的会话信息确定是否存在存内泄漏（即资源没有正常释放，导致会话属性一直增加）。

节点	属性	描述
Sessions	Count	会话总数。
	maxMemory	JVM最大可以申请的内存数。
	totalMemory	JVM当前已经申请的内存数。
	freeMemory	JVM当前空闲的内存数。
Session（会话）	id	会话ID。
	maxInactiveInterval	最大允许空闲的时间（秒）。
	ip	最后访问的客户端IP地址。
	user	Smartbi登录用户名，NotLogin表示未登录。
Item（属性）	name	属性名称。
	className	属性类型。

## listpools.jsp（对象池信息）

对象缓冲池可以增强系统在并发时的性能，减少服务器的压力，提高用户报表查询速度。

节点	描述
JVM BusinessViewBOMap	JVM中数据集定义对象缓冲池 。 Size: 数据集定义对象个数 。 Values: 内存地址+数据集定义名称列表。
JVM CellElementMap	JVM中复杂报表单元格对象缓冲池 。 Size: 复杂报表单元格对象个数 。 Values: CE[532, 1, 1, 1]厦门。
JVM DBSQLResultStoreMap	JVM中业务数据缓冲池 。 Size: 业务数据对象个数 。 Values: 业务数据对象基本信息，如：数据集ID、参数等。
FixedReportPoolCacheMap	JVM业务报表对象缓冲池 。 Size: 业务报表对象个数。
ExecutedReportPoolCacheMap	JVM复杂报表运行时对象缓冲池 。 Size: 复杂报表运行时对象个数 。

FixedReportPool	业务报表对象池，用于缓存复杂报表表样对象，减少重复创建次数。 Active：当前活动的业务报表对象个数。 Idle：已释放的业务报表对象个数。
ExecutedReportPool	复杂报表运行时对象池，用于缓存复杂报表的数据对象，减少重复访问数据库获取数据。复杂报表在没有设置回写规则的情形下，才存在复杂报表运行时对象池。 Active：当前活动的复杂报表运行时对象个数。 Idle：已释放的复杂报表运行时对象个数。
BusinessViewBOPool	数据集定义对象池，用于缓存报表所依赖的数据集对象，减少重复创建次数。 Active：当前活动的数据集定义对象个数。 Idle：已释放的数据集定义对象个数。
DBSQLResultStorePool	业务数据对象缓存池，根据报表中的条件（如参数、排序等）缓存业务数据，减少重复访问数据库获取数据。 Active：当前活动的业务数据对象个数。 Idle：已释放的业务数据对象个数。
BaseBofKeyedPoolableObjectFactory	BOF_OBJECT_POOL_CACHE size：当前所有的对象连接池的总的Key个数（包括本机直接创建的对象和其他服务器创建通知同步过来的）。 CacheIdentity2KeyMap size：仅仅只包含由本服务器创建的对对象的Key集合。
ConnectionPool	数据库连接池（res为知识库）。 Name：数据源名称。 Active：当前活动的连接个数。 Idle：已关闭连接个数。

## getclassurl.jsp（查找加载类路径）

用于查看虚拟机当前加载的类，对应的文件路径，从而方便定位解决“类”冲突的问题。通常用于查看JDBC驱动加载冲突的情况。

节点	描述
classname	类全名。
加载类的文件路径	显示该类实际在Smartbi服务器中存放的位置，通常会有bof-Common.jar!/bof/BOFException.class，其中“!”表示bof-Common.jar为一个压缩文件。
类加载器	显示加载该类的类加载器及其祖先类加载器，若显示为“Bootstrap加载器”则表示该类是由JVM的Bootstrap加载器加载的。

## listwsdl.jsp（WSDL地址）

列出Smartbi中内置的WebService API接口，并提供这些WebService的WSDL访问地址。用于二次开发。

## threaddump.jsp（线程堆栈）

用于定时将JVM所有的线程堆栈打印到临时目录中，方便定位服务器运行缓慢、假死等情况。访问该页面在打印间隔中输入合适的值（默认是10秒）后，点击“开始打印”。页面会显示“线程堆栈日志目录”，在该目录中会定时生成一个新的堆栈信息文件。通过直接访问服务器上的文件系统或者通过登录Smartbi服务器点击“管理”“工具”“导出系统日志”可以将这些信息文件全部导出。

## jsconsole.js（JS控制台）

IE8以上的浏览器版本默认自带有JS控制台，IE8之前的版可以使用Smartbi系统中的JS控制台。JS控制台可以方便用户调试一些脚本错误。其中一些使用技巧：

1．如果不是在Eclipse的开发环境中使用此功能，在打开该控制台后，此后从服务器返回的JavaScript不会使用JSMIn进行压缩，但是原有的JavaScript会保持压缩状态。因此需要配合Charles中的“No Caching…”功能并刷新IE使得原有JavaScript不压缩方便调试。

2．需要在指定方法中增加调试的方法（以QueryView.doExport增加断点为例）：

```
1) 在控制台输入 var QueryView = jsloader.resolve("freequery.query.QueryView");if(!QueryView.prototype._doExport) { QueryView.prototype.doExport = QueryView.prototype.doExport; QueryView.prototype.doExport = function(ev) { debugger; this._doExport(ev); }}。
```

2) 点击“Run”运行该脚本，注意：刷新IE会导致该脚本失效需要重新运行。

3) 在下次创建的QueryView实例调用doExport方法时就会触发debugger语句进入调试状态。

## heapdump.jsp（JVM堆栈）

该页面显示目前JVM堆中所有对象个数及占用的内存数，此功能需要Smartbi服务器运行在Sun的JDK1.6以上版本，否则无法正常使用。class name是这些对象的类型，其中“[L”前缀是表示数组的意思。通常情况下是JDK基础类占用最多，这些通常可以忽略。应该着重关注bof和com.freequery包名下的类。

“下载HeapDump.bin”链接是将JVM当前堆全部保存到文件中然后下载到客户端。该功能需要花费比较多的时间而且下载的文件非常大，下载后的文件可以使用JVisualVM打开分析（参考本文档JVisualVM的说明）。

## CPUMonitor.jsp（CPU时间）

以图形方式显示当前JVM所占用的CPU时间。

## Memory.jsp（内存）

以图形方式显示当前JVM所占用的内存。在该图形中Y轴的最大值表示服务器最大可以申请的内存（maxMemory），黄色线表示当前已经申请的内存（totalMemory），蓝色线表示当前已经使用的内存（totalMemory - freeMemory）。

## listExtensions.jsp（加载扩展包列表）

列举当前Smartbi中已经加载的扩展包。当Smartbi加载压缩为ext后缀的扩展包时，会将该扩展包解压缩到系统临时目录中，所以“加载路径”会是系统的临时目录。

## listobjects.jsp（系统对象个数列表）

该页面会遍历Smartbi应用中的所有类及依赖类库，如果该类已经加载则遍历它的属性找出所有引用关系。该页面主要作用查找是否有一些static或者list等没有释放的问题。一般情况不需要使用此页面。

## testmulticast.jsp（分组广播测试）

用于测试需要部署Smartbi集群的服务器网络环境是否支持分组广播。Smartbi在集群中需要使用EhCache中的广播机制通知其它服务器，但是在某些网络环境中分组广播可能被禁止，该页面是提供测试使用的。其中的参数意义：

- 分组地址——分组广播的地址，必须在224.0.0.0~239.255.255.255之间，不建议修改默认值。
- 端口—— 分组广播的端口号，必须为合法端口号（1~32766），一般不需要修改。但是如果两套Smartbi集群环境同时运行，则端口号不能相同。
- TTL—— 路由数目，同一子网内应该是1，每经过一个路由需要增加1（具体是多少请联系网络管理员）。但是在比较多的网络环境中分组广播都是不允许跨子网的，所以跨子网的分组广播可能会失败。

测试过程中需要启动两个IE并分别访问两台服务器（同一个服务器上测试没有意义），在“分组地址”、“端口”和“TTL”中输入相同的参数值。然后其中一台服务器中先点击“接收”按钮，然后另外一台服务器点击“发送”（注意“接收”是10秒超时的）。

发送的服务器应该会显示“发送成功！”，接收的服务器应该会显示“接收到：bof.Multicast.test！”，这个表示从发送到接收服务器的分组广播测试成功。但是注意还必须反过来测试是否成功，因此有一些网络环境可能会出现A机器可以向B机器发送成功，但是反之失败的情况。

若接收服务器一直接收不到信息，则会在10秒后显示“java.net.SocketTimeoutException:Receive timed out”，这个表示无法正常传递分组广播。这时Smartbi的集群就需要“peerDiscovery>manual”的配置（具体请参考Smartbi集群配置文档）。