# 生产运维文档

本文档为网状网日常生产运维各个方面的总结，包括操作步骤与检查方法以及出错后的处理等等，仅供网状网生产运维人员使用。

# 生产部署上线

## 1.1上线前（秀娟）

一般情况下，网状网每月一次大版本上线，上线之前，会从需求组得到上线的文档，文档中包含上线的配置内容与上线时间，同时需要询问BBOSS项目组与一级客服项目组，问他们需要网状网需要配合上线的内容。在上线之前，需要明确以下几个方面：

1. 新上线平台的domain，从哪里接入，我们给分配的端口是那个（端口分配工作目前由秀娟与刘畅共同完成），新上线平台是否有落地业务，落地的URL地址是什么，网络是否已经打通(上发网络也需要同时打通)，网络打通工作由刘畅配合平台共同完成。
2. 新上线业务的bip与activity，测试环境是否测试完成(曲悦负责)
3. ftp需要配合的内容，由李晴负责，ftp有专门的运维文档，请参考

以上确认完成后，就可以进行生产的上线

## 1.2生产上线过程

### 1.2.1 数据库配置（秀娟）

1. 曲悦根据上线内容在测试环境提取相应上线的sql语句以及xsd
2. 秀娟配置生产环境(10.1.128.136,mc用户)的以下表格：

如果是T上线，配置t\_t\_activity, t\_t\_bip, t\_t\_bip\_activity, t\_t\_activity\_action, t\_t\_domain, t\_t\_party, t\_t\_party\_bip, t\_t\_party\_routing

如果是S上线，配置s\_t\_activity, s\_t\_bip, s\_t\_bip\_activity, s\_t\_activity\_action, s\_t\_domain, s\_t\_party, s\_t\_party\_bip, s\_t\_party\_routing

主要按照测试环境的sql语句，执行过程中需要检查上线内容是否符合，插入数据库的时间都写成sysdate，party\_bip注意每一个业务需要两条相应的记录（生产一条，测试一条），两个需要特殊注意的表格：

* party表，对于T模块，如果没有特殊的路由，party表中的local\_req\_url应该配置成LOCALHOST，配置成localhost后，路由方式为：不是同duns的走正常配置路由，同duns的直接发送到party\_routing表中配置的地址，如果对于同duns的有特殊配置，将地址写入到local\_req\_url列，同duns的将发送到此地址，此地址相当于同duns的hsn。

对于S模块，如果没有特殊的路由，party表中的local\_req\_url应该配置成0。同duns的时候不做特殊配置，则写0，走路由方式，寻找hsn。如果local\_req\_url指定地址，则同duns的将发送到此地址，此地址相当于同duns的hsn（S模块对于同duns没有直接发送落地方的功能）。

* party\_routing表，测试环境过来的地址都是测试的地址，在生产环境不适用，如果是省BOSS落地，需要使用party\_routing.sql文档中的地址，替换一下activity\_id,如果是新接入平台落地，需要替换成平台提供的地址，如果是就平台新业务，需要替换成生产上已经存在的平台地址。对于信息港proxy出口的地址配置格式为：<http://10.255.242.71:1132/TransProxy/out?to=http://112.33.0.187:4004/WepttBossAdaptation/mobile/bossAdaptation.action（to>后面的部分为实际平台落地地址）；博瑞棋F5出口的，需要配置成平台在F5的地址，目前博睿琦已经不再接入新的平台，新平台外网接入都从redware接入。

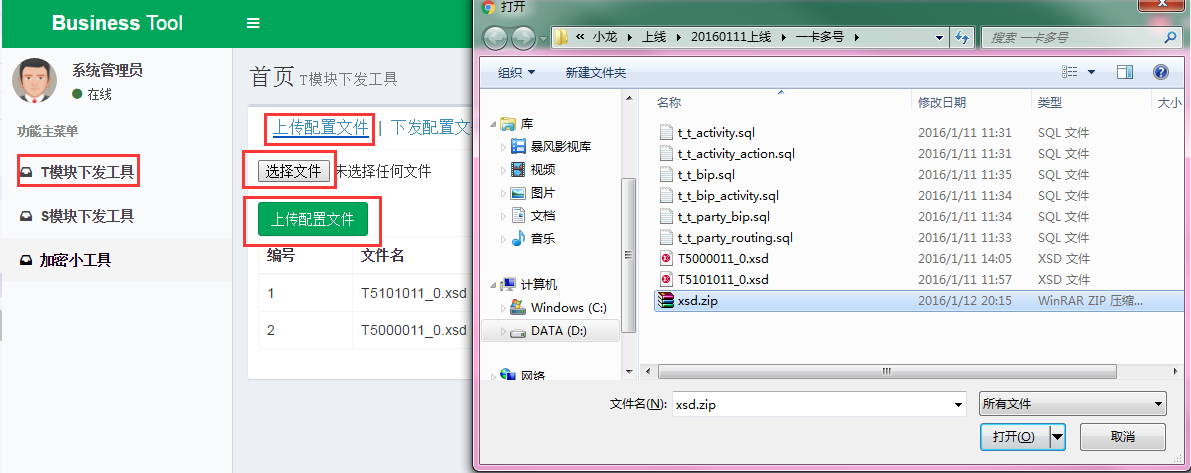
### 1.2.2 xsd上传（小龙）

新版的xsd上传工具由杨依开发，上传过程不明白的可以询问小龙，中间程序问题可随时找杨依解决。

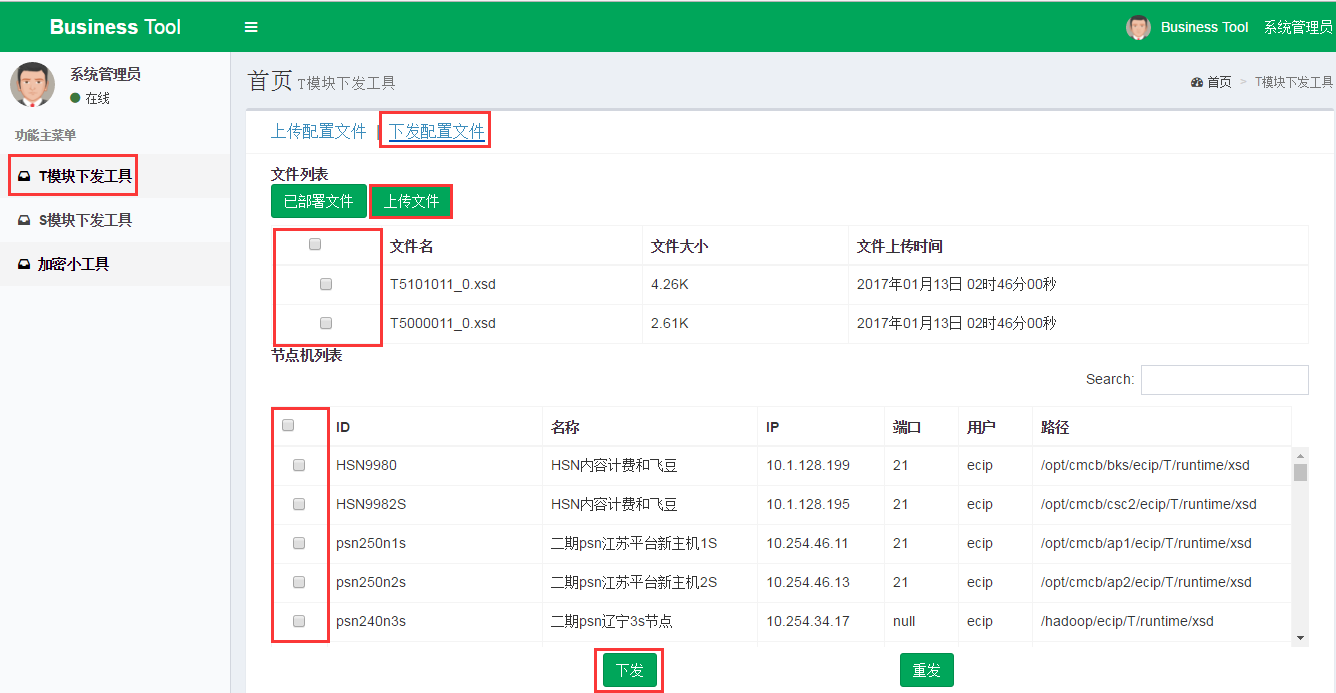
1. 生产环境浏览器设置10.1.128.200:9000的代理，访问xsd上传地址：<http://10.1.128.112:8090/BusinessTool/manage/main.html>，用户名：admin，密码：123456。登录后的界面如图：



1. 根据上传的xsd选择左边的模块（T或者S），点击界面中“选择文件”按钮，在弹出的窗口中选中需要上传的xsd压缩包，点击“打开”按钮，随后点击“上传配置文件”按钮。



1. 点击“上传配置文件”按钮后，再点击“上传文件”按钮，勾选需要上传的xsd文件后，再勾选需要上传的省或者平台节点机，最后点击“上传”按钮，等待上传显示成功即可，如有上传不成功的，点击“重发”按钮直到成功，少量没有上传成功的，可自行手动通过SFTP或者SCP方式上传即可。如果没有上传成功是由于机器的密码不对或者机器路径配置问题，参见1.2.3节check一下数据库里面的配置，确保数据库配置没有问题。

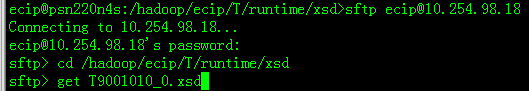


**补充：手动上传的SFTP方式与SCP方式**

**手动上传的SFTP方式**

1. 登录上传xsd失败节点机上，进入xsd目录；
2. 通过sftp ecip@IP（该IP为上传xsd成功的节点机IP）命令以sftp方式进入上传xsd的机器上；
3. 通过get xs。d名称来获取xsd即可。

以天津（10.254.14.18）上传xsd失败，云南（10.254.98.18）上传成功为例，如图通过sftp方式手动获取T9001010\_0.xsd文件。

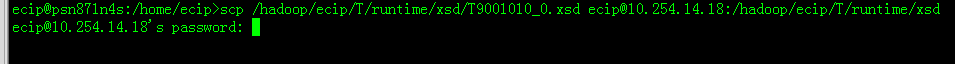


**手动上传的SCP方式**

1. 登录上传xsd成功节点机上；
2. 通过命令以scp方式向天津省上传xsd文件，命令如下：

scp xsd文件 ecip@IP（上传xsd失败的节点机IP）：失败节点机xsd目录

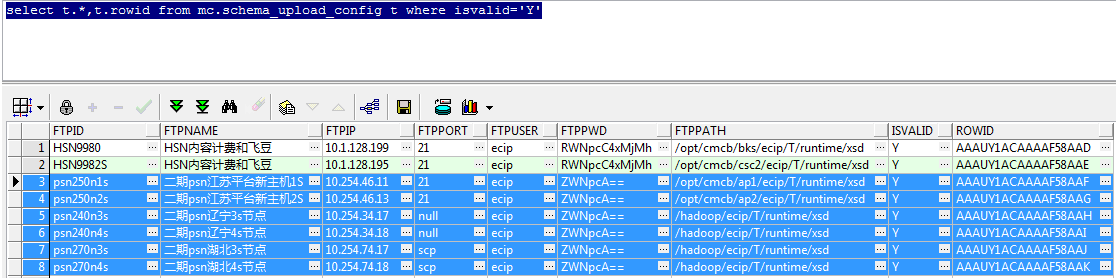
以天津（10.254.14.18）上传xsd失败，云南（10.254.98.18）上传成功为例，如图通过scp方式手动获取T9001010\_0.xsd文件，输入密码即可从云南省上传xsd到天津省。



### 1.2.3 xsd工具补充（小龙）

* **xsd新增上传节点**

1.登录4A上的136数据库，找到xsd新增上传节点的数据库配置表mc.schema\_upload\_config，通过select t.\*,t.rowid from mc.schema\_upload\_config t方式插入需要新增的节点信息。如下：



表字段说明：

FTPID：该字段为FTP节点id，用于区分省上传的各个节点机。建议通过hostname来获取节点机名称用以命名FTPID。

FTPNAME：该字段说明节点机名称详细信息。

FTPIP：该字段为新增上传节点的机器IP。

FTPPORT：该字段是区分不同的上传xsd方式。21：FTP方式，null：SFTP方式，scp：scp方式。在新增节点的机器上通过uname命令来获取机器是HP-UX还是Linux系统，如果是HP-UX系统，建议用ftp方式；如果是Linux系统，优先使用SFTP方式，sftp方式上传xsd时，多次上传xsd失败修改为scp方式。

FTPUSER：该字段默认为ecip用户。

FTPWD：该字段为ecip用户密码，这个密码为密文形式。各个ecip密码密文如下：ecip的密文是ZWNpcA==，Ecip.123！的密文是RWNpcC4xMjMh，Um\_yH%1lD的密文是VW1feUglMWxE。

FTPPATH：该字段是新增节点的机器上xsd目录。

Isvalid：该字段用于判断是否向该节点上传xsd。

2.刷新<http://10.1.128.112:8090/BusinessTool/manage/main.html>网址界面即可实时看到新增的节点，随后勾选就可以上传。

* **xsd工具程序重启**

1.登录10.1.128.112机器，ps –exf | grep tomcat找到tomcat进程；

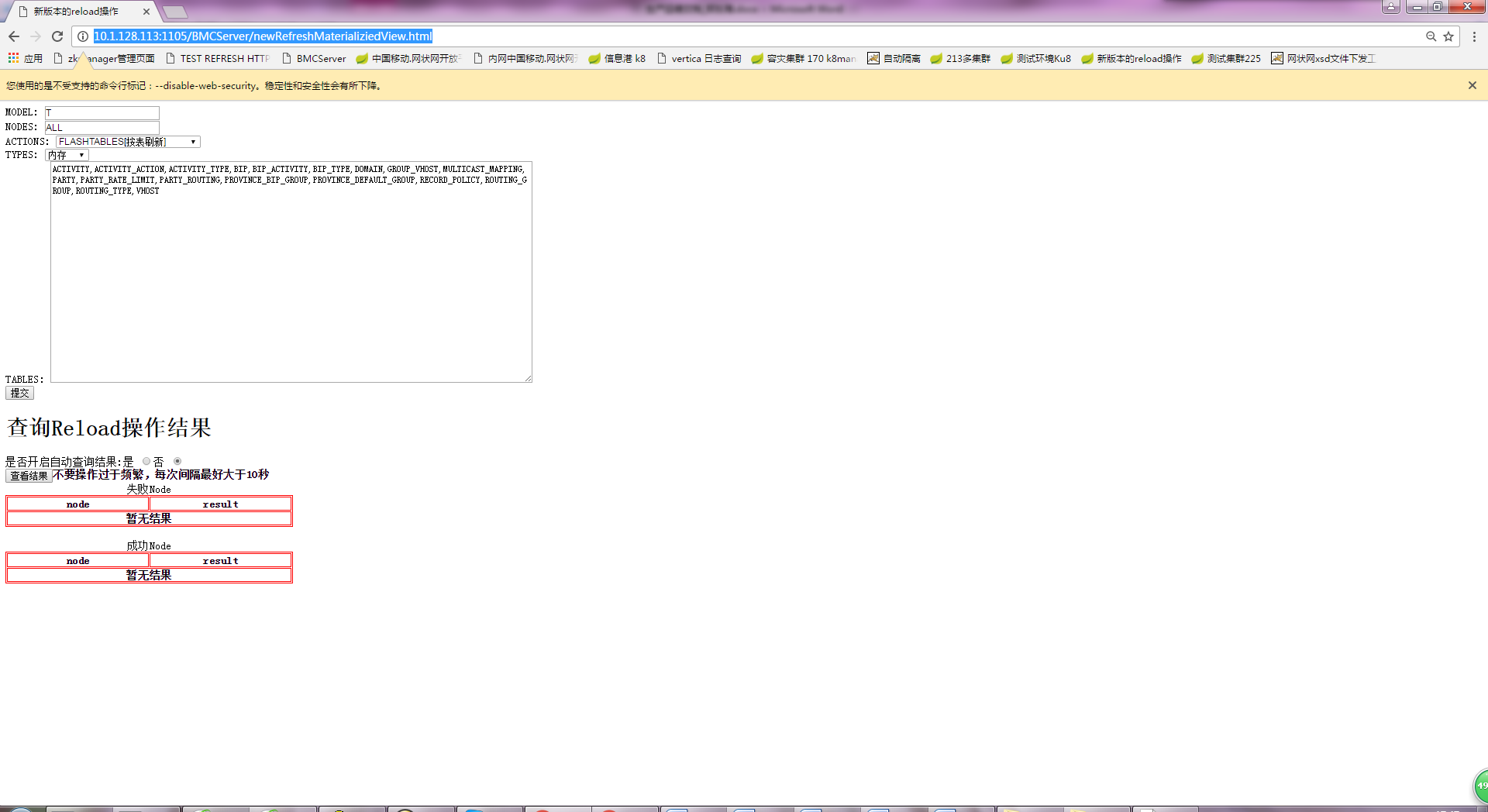
2.通过 “kill -9 进程号”命令杀掉tomcat进程；

3.进入/opt/ecip/portal/tomcat/bin目录，执行startup.sh命令启动tomcat即可。如图



### 1.2.4数据库表与内存刷新（东海）

#### 1、谷歌浏览器登陆http://10.1.128.113:1105/BMCServer/newRefreshMaterializiedView.html



#### 2、参数值配置

MODULES：选择模块，目前支持T和S。同一个页面只能支持一个模块。

NODES：

说明：

zookeeper中配置功能的exe\_para中的node id确认操作目标，也可以理解为zkagent名字。表示需要哪个zkagent去执行你下、面想要执行的操作。如下图：是通过访问zkmanager：http://10.1.128.113:1105/zkmanager/

通过选择reload参数查看的exe\_para内容。Node=zkagent名字。该node下的url就是你即将实际要操作的T或S应用的地址。

取值：

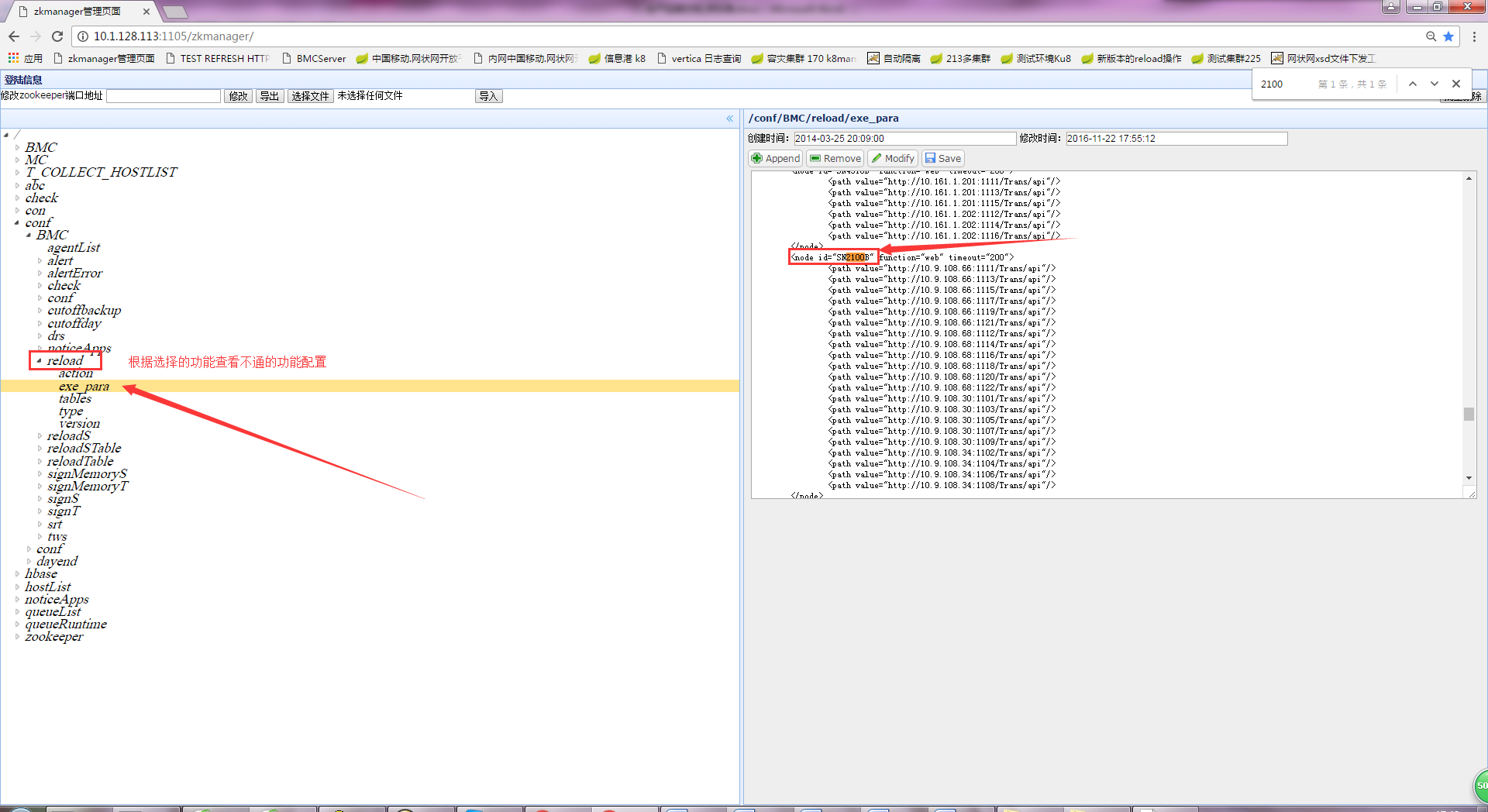
ALL表示所用zkagent都执行操作

多个node值用英文逗号隔开：SN1000B,SN9980N,SN2000N

一个node直接写即可。

注意： 关键是需要通过zkmanger查看你要知道有哪些名字。请通过zkmanager查看对应的功能的exe\_para内容。

如下是reload的zkmanager查看exe\_para的内容：



ACTIONS:

FLASHTABLES[按表刷新]:

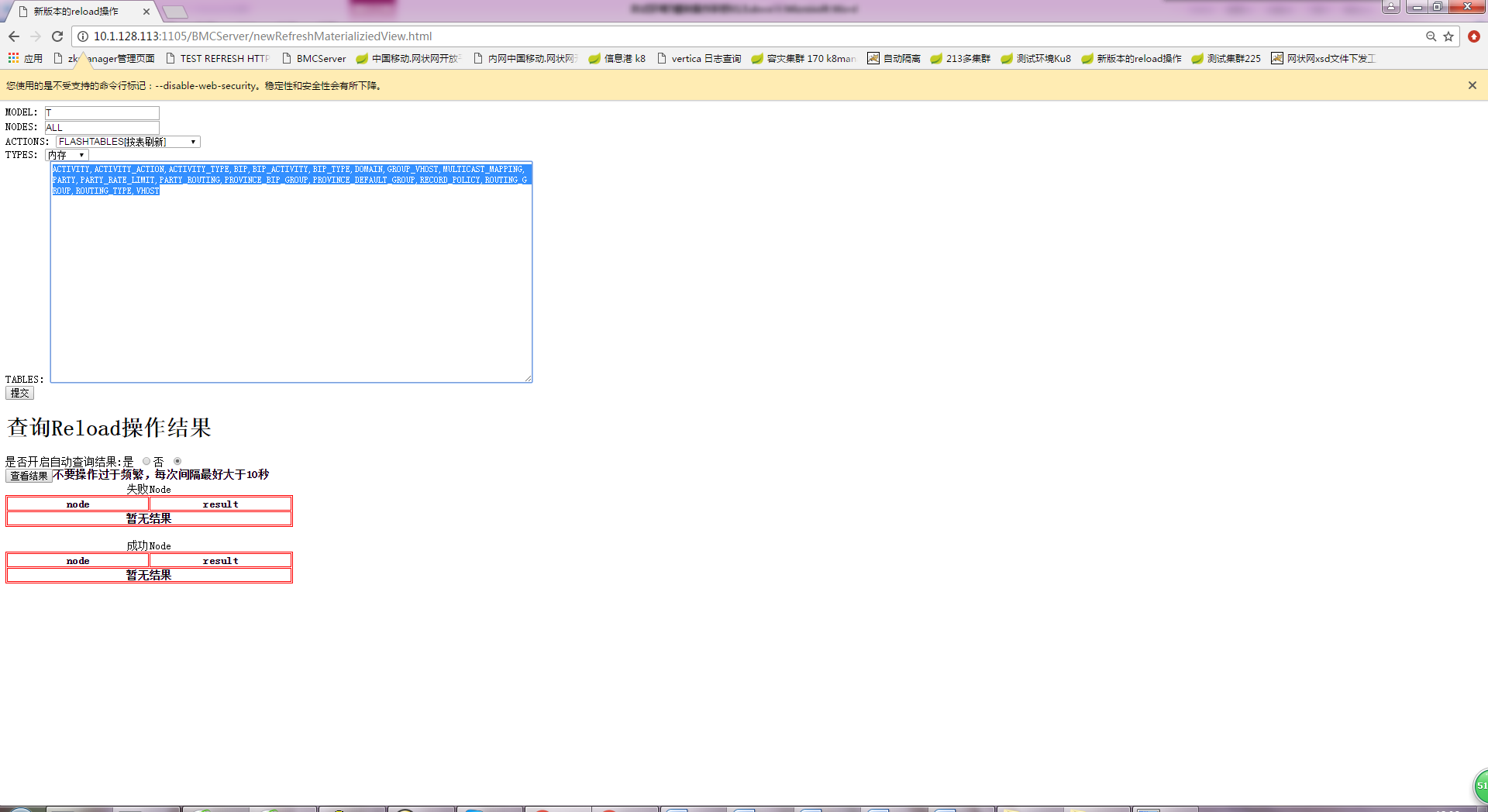
当选择该选项时，意味着需要刷新TABLES里面的物化视图.所有物化视图都可以在table中填写，使用逗号分开。如下是可以填写的物化视图：(包括PARTY\_BIP)

ACTIVITY,ACTIVITY\_ACTION,ACTIVITY\_TYPE,BIP,BIP\_ACTIVITY,BIP\_TYPE,DOMAIN,GROUP\_VHOST,MULTICAST\_MAPPING,PARTY,PARTY\_RATE\_LIMIT,PARTY\_ROUTING,PROVINCE\_BIP\_GROUP,PROVINCE\_DEFAULT\_GROUP,RECORD\_POLICY,ROUTING\_GROUP,ROUTING\_TYPE,VHOST,PARTY\_BIP

注意：

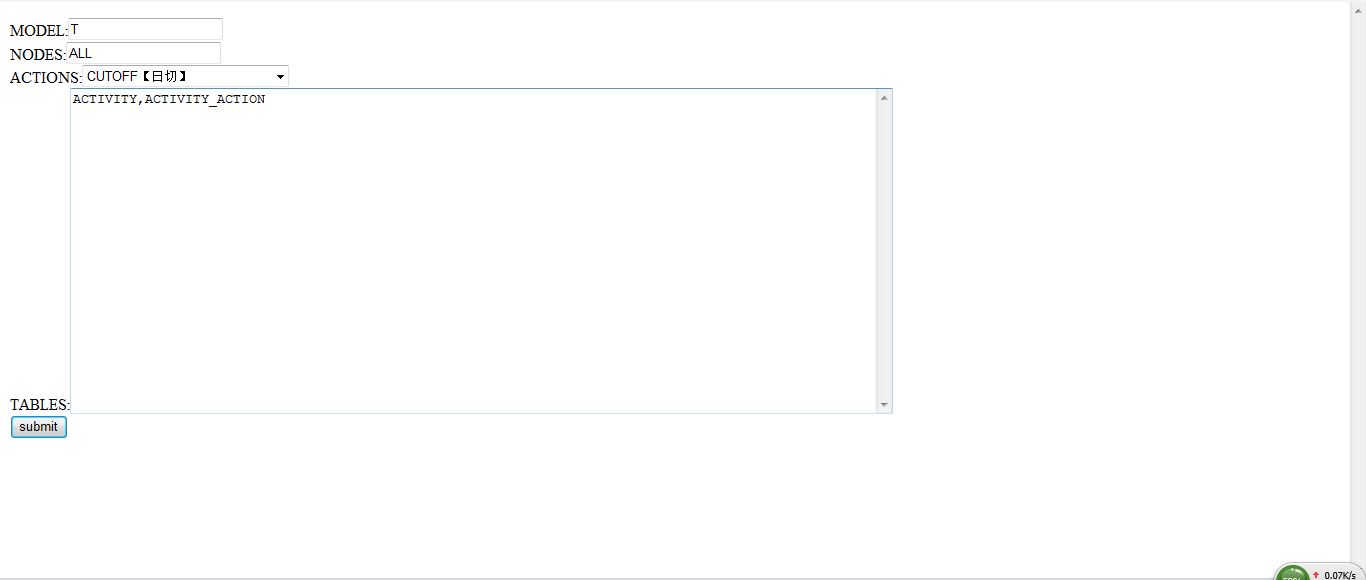
1. 数据库表刷新的时候，选择的表名越精确越好，不要有冗余的表，程序会按照选择的表名进行数据库表的刷新物化视图操作
2. 所有表的刷新都使用FLASHTABLES action。
3. party\_bip表需要单独进行刷新，不要跟那些小表混在一起做。

如下图：



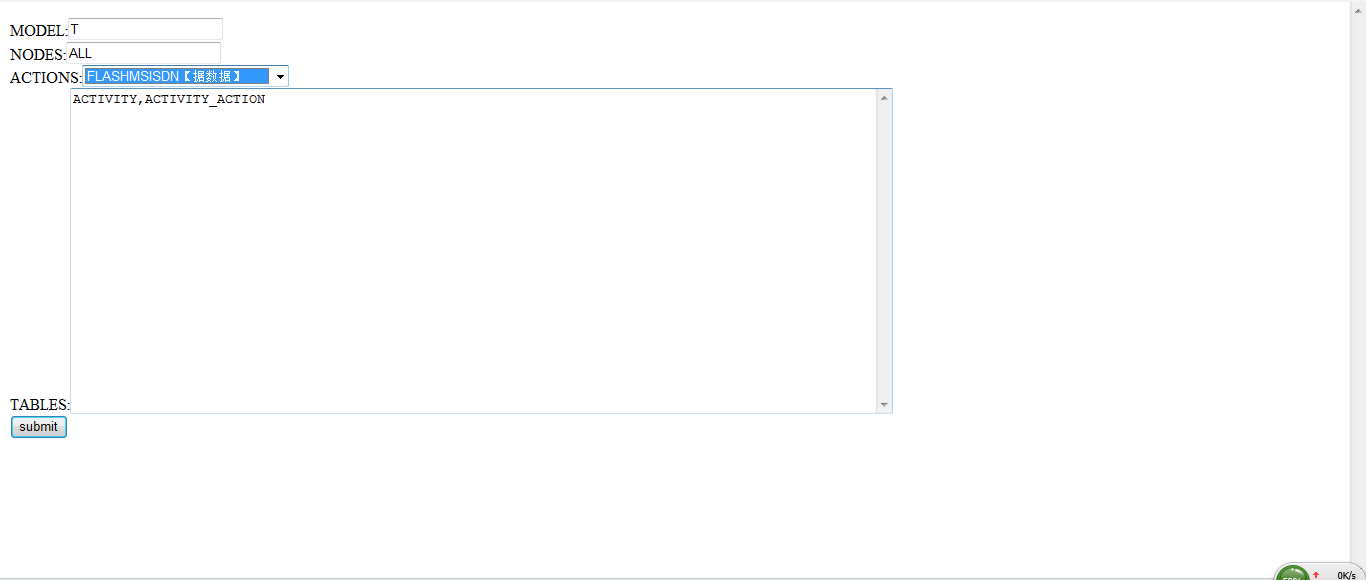
CUTOFF【日切】：(目前日切的补救措施主要使用网状网能力开放系统操作)

此选项代表刷，新日切sys\_para表，一般日切出现异常的时使用该选项。TABLES里面的值不为空即可（该值此时无效）。如下图：



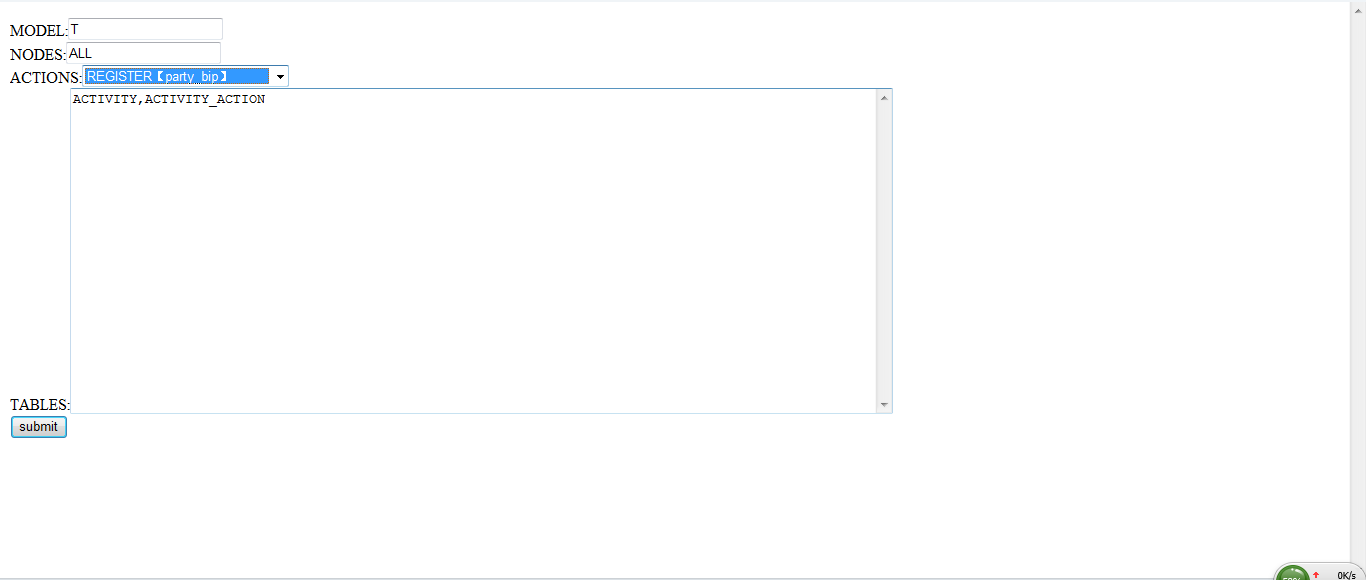
FLASHMSISDN【据数据】：

选择该选项，意味着刷新局数据MSISDN\_AREA\_MAP表。只有局数据出现更新时才执行该选项。TABLES里面的值不为空即可（该值此时无效）。如下图：



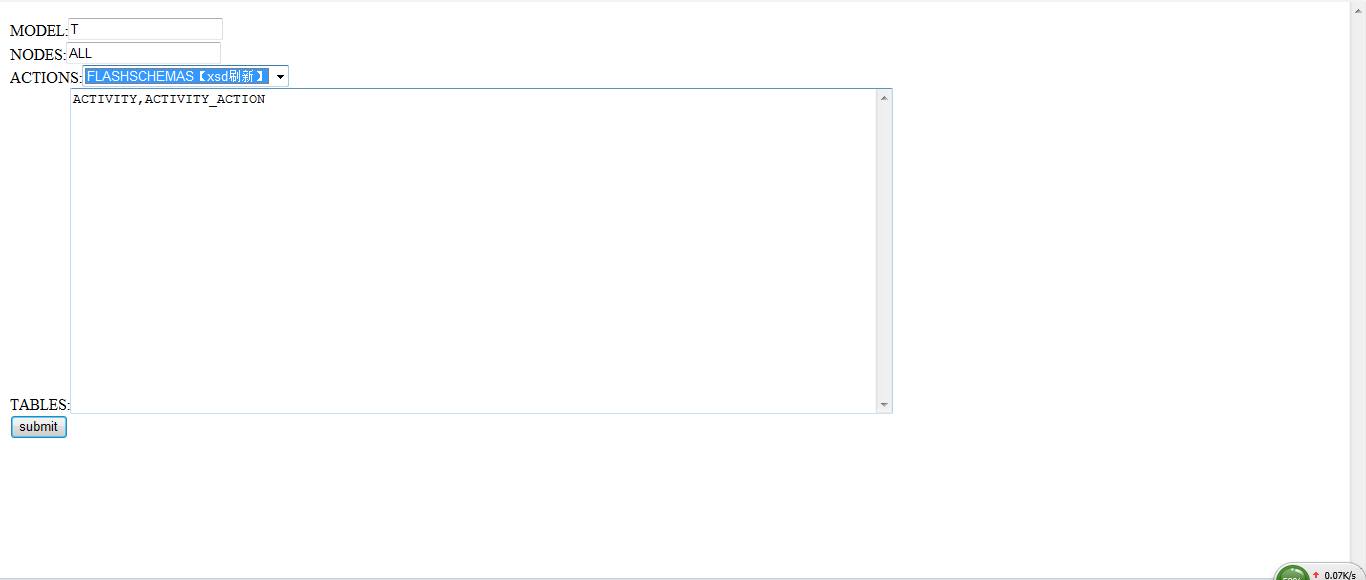
REGISTER【party\_bip】

选择该选项，意味着刷新PARTY\_BIP表。只有该表需要刷新内存时执行该选项。TABLES里面的值不为空即可（该值此时无效）。如下图：



FLASHSCHEMAS【xsd刷新】

选择该选项，意味着刷新xsd校验文件。只有校验文件有改变时需要刷新内存时才执行该选项。TABLES里面的值不为空即可（该值此时无效）。如下图：



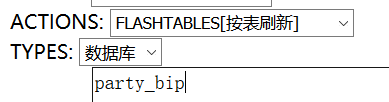
#### 3、选择action操作类型

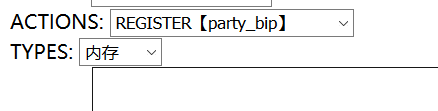
分为“内存”和“数据库”两种。

数据库刷新主要是action为FLASHTABLES时使用，这里表示所有分库从MC数据库刷新数据。

内存刷新表示将表数据捞到T或S应用内存中。

注意：party\_bip表做数据库刷新的时候，使用按表刷新的方法，如下图：



Party\_bip表内存刷新的时候，使用party\_bip自带的内存刷新方法，如下图

#### 4、页面查看操作结果

勾选“否”时，手工点击查看结果查看实时结果。勾选“是”时，页面自动按照指定的秒数间隔，自动刷新结果。等待300秒后最终结果。



### 1.2.5 刷新数据库内存过程中的注意事项 （东海）

##### 1.2.5.1 刷新注意选择

表刷新时，选择FLASHTABLES。内存刷新时请选择对应的ACTION进行操作。

##### 1.2.5.2界面操作反馈结果错误处理方法

请按照如下流程检查相关日志：BMCServer->zkagent->T,S应用或者BMCServer->zkagent5.1->scmp->T,S容器。根据日志描述定位问题。Zkagent 地址请参考表格“zkagent&db info.xlsx”。

正确日志应该如下：

###### BMCServer日志检查

会收到类似:

[INFO][17:42:54:499][com.hp.ecip.bmc.service.BmcService.newReload(85)] >> ==>> reload: ALL == PROVINCE\_DEFAULT\_GROUP,ROUTING\_GROUP,VHOST,GROUP\_VHOST == database ---操作内容

如果错误，请重启BMCServer。 登陆http://10.1.128.113:1900/，admin/admin,重启节点aloneIns2（BMCServer）。也可以登录113杀掉aloneIns2应用(instance)，让其自动重启。

###### Zkagent日志检查

0116 17:43:10.938[com.hp.ecip.zk.notice.NtcResp$2(138)] INFO||SN2000N well---zk data change-->/noticeApps/reloadTable ---action触发

0116 17:43:10.948[com.hp.ecip.zk.notice.NtcResp$2(144)] INFO||SN2000N hostList : ALL:SNCNTT

0116 17:43:10.948[com.hp.ecip.zk.notice.NtcResp$2(147)] INFO||SN2000N begin job do thread invoke..SNCNTT

0116 17:43:10.968[com.hp.ecip.zk.notice.ZkCall(67)] INFO||SN2000N exe para : nodeId:SN2000N function:web path.size1 ----zkagent读到的操作url个数

0116 17:43:10.969[com.hp.ecip.zk.notice.ZkCall(75)] INFO||=>> use para timeout num:200

0116 17:43:13.101[com.hp.ecip.zk.notice.ZkCall(127)] INFO||1 <==> SN2000N Do http.. [1] ==>http://10.255.242.20:1101/Trans/api version=20170116174255204&tables=PROVINCE\_DEFAULT\_GROUP,ROUTING\_GROUP,VHOST,GROUP\_VHOST&type=database&action=

FLASHTABLES <<== Result:1 ---**http发起参数：请对应刷新表的页面参数：version：时间戳，每次action的区分标识。**

**type:database和memory两种，分别对应表和内存刷新。**

**tables: 操作的表，只有表刷新**FLASHTABLES action才有用。其他时候不需要关注。

action: 界面ACTIONS下拉列表中的一种。具体值请参考本章节“参数值配置”说明。

Result： 0 表示失败，1表示成功。如果出现参数不对，请检查页面操作或者exe\_para配置。

0116 17:43:13.102[com.hp.ecip.zk.notice.ZkCall(130)] INFO||SN2000N execute http://10.255.242.20:1101/Trans/api ok..

0116 17:43:13.137[com.hp.ecip.zk.notice.ZkCall(305)] INFO||SN2000N set result[1]/noticeApps/reloadTable/SNCNTT/result/SN2000N --zkagent写入结果到zookeeper，有时候会出现nodeExisits报错，请重启zkagent

0116 17:43:13.137[com.hp.ecip.zk.notice.ZkCall(176)] INFO||ZK Call execute job all over..

如果发现zookeeper配置（参考zookeeper配置部分）中配置了url，但是zkagent并没有触发http请求，请重启zkagent。重启方法参考本文档“重启agent”部分。

###### Scmp日志检查（假设是集群）

Scmp作为集群的代理，转发zkagent转发过来的所有请求（比如zkagent的http请求）到集群中的T和S节点，将来可能会包含其他模块，后续可能会有新的需求更新，目前是如此。Scmp一般在集群的master主机上。可以执行命令查找scmp部署路径：find / -name scmp (unix,linux),

locate –b scmp (linux)

查找。查看scmp日志：scmp的相对简单，只要刷新了如果报错，请进容器查看应用日志。

\*/scmp/tomcat-7/logs/kuber.log

[com.hp.kubernetesInterface.service.impl.ExeKubeBusiness.exePost(BusinessActionImpl.java:756)INFO ,2017-01-16 18:43:58.294]execute kubebusiness start........

[com.hp.kubernetesInterface.service.impl.BusinessActionImpl.executeHttpRequestWithReadTimeByThreads(BusinessActionImpl.java:343)INFO ,2017-01-16 18:43:58.295]begin run Threads ..

[com.hp.kubernetesInterface.service.impl.ExeHttpWithReadTimeThread.call(BusinessActionImpl.java:582)INFO ,2017-01-16 18:43:58.295]{tables=ACTIVITY,ACTIVITY\_ACTION,ACTIVITY\_TYPE,BIP,BIP\_ACTIVITY,BIP\_TYPE,DOMAIN,GROUP\_VHOST,MULTICAST\_MAPPING,PARTY,PARTY\_RATE\_LIMIT,PARTY\_ROUTING,PROVINCE\_BIP\_GROUP,PROVINCE\_DEFAULT\_GROUP,RECORD\_POLICY,ROUTING\_GROUP,ROUTING\_TYPE,VHOST, action=FLASHTABLES, type=memory, version=20170116184333360} --请求参数

[com.hp.kubernetesInterface.service.impl.ExeHttpWithReadTimeThread.call(BusinessActionImpl.java:582)INFO ,2017-01-16 18:43:58.296]{tables=ACTIVITY,ACTIVITY\_ACTION,ACTIVITY\_TYPE,BIP,BIP\_ACTIVITY,BIP\_TYPE,DOMAIN,GROUP\_VHOST,MULTICAST\_MAPPING,PARTY,PARTY\_RATE\_LIMIT,PARTY\_ROUTING,PROVINCE\_BIP\_GROUP,PROVINCE\_DEFAULT\_GROUP,RECORD\_POLICY,ROUTING\_GROUP,ROUTING\_TYPE,VHOST, action=FLASHTABLES, type=memory, version=20170116184333360}

… [com.hp.kubernetesInterface.service.impl.BusinessActionImpl.executeHttpRequestWithReadTimeByThreads(BusinessActionImpl.java:346)INFO ,2017-01-16 18:44:20.054]Threads run end..

[com.hp.kubernetesInterface.service.impl.ExeKubeBusiness.exePost(BusinessActionImpl.java:776)INFO ,2017-01-16 18:44:20.054]execute paras is [tables = ACTIVITY,ACTIVITY\_ACTION,ACTIVITY\_TYPE,BIP,BIP\_ACTIVITY,BIP\_TYPE,DOMAIN,GROUP\_VHOST,MULTICAST\_MAPPING,PARTY,PARTY\_RATE\_LIMIT,PARTY\_ROUTING,PROVINCE\_BIP\_GROUP,PROVINCE\_DEFAULT\_GROUP,RECORD\_POLICY,ROUTING\_GROUP,ROUTING\_TYPE,VHOST&action = FLASHTABLES&type = memory&version = 20170116184333360]

[com.hp.kubernetesInterface.service.impl.ExeKubeBusiness.exePost(BusinessActionImpl.java:777)INFO ,2017-01-16 18:44:20.054]execute pod number [102] –http请求个数

[com.hp.kubernetesInterface.service.impl.ExeKubeBusiness.exePost(BusinessActionImpl.java:778)INFO ,2017-01-16 18:44:20.055]execute kubebusiness finished!

发现scmp并没有给容器发送请求，请重启scmp所在的tomcat。ps aux|grep scmp 获取进程号，然后kill掉进程。最后进入到tomcat目录/bin/ 执行./start\_up.sh 命令。启动成功后请查看scmp日志，如果一致有日志输出且ps进程成功表示启动完成。

###### T日志检查

容器和实体应用都是T的应用，日志内容是一样，如下是容器收到表内存刷新日志：

红色高亮部分为收到的参数，参数意思和http请求参数意义一样。

实体应用请查看server.log.

容器请查看：/var/run/tomcat/logs/catalina.log

grep -E 'API|was loaded' server.log 或者 grep -E 'API|was loaded' catalina.log

结果如下：（参数意义不再赘述）

####|2017-01-16 17:43:20.438|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,438 INFO >>> TransAPI request action original type : database

####|2017-01-16 17:43:20.438|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,438 INFO >>> TransAPI request action version : 20170116174255204

####|2017-01-16 17:43:20.439|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,439 INFO >>> TransAPI request action cutoffday : null

####|2017-01-16 17:43:20.449|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,449 INFO >>> TransAPI request action dataType : null

####|2017-01-16 17:43:20.449|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,449 INFO >>> TransAPI request action real type : database

####|2017-01-16 17:43:20.449|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,449 INFO >>> TransAPI processing FLASHTABLES request.

####|2017-01-16 17:43:20.449|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,449 INFO >>> TransAPIHandler doFlashtables now.

####|2017-01-16 17:43:20.450|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,450 INFO >>> TransAPIHandler doFlashtables transApiActionType is database.

####|2017-01-16 17:43:20.465|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,465 INFO >>> TransAPIHandler existVersion [0] --检查是否已经操作过 0表示没有，1表示已经操作。如果是1则直接反馈成功。0则继续操作。

####|2017-01-16 17:43:20.474|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,474 INFO >>> TransAPIHandler doFlashtables version not exist.

####|2017-01-16 17:43:20.474|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,474 INFO >>> TransAPIHandler refreshView [PROVINCE\_DEFAULT\_GROUP] Before get connetion.

####|2017-01-16 17:43:20.476|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,476 INFO >>> TransAPIHandler refreshView [PROVINCE\_DEFAULT\_GROUP] Get connetion done. –获取数据库连接

####|2017-01-16 17:43:20.476|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,476 INFO >>> TransAPIHandler refreshView [PROVINCE\_DEFAULT\_GROUP] PrepareCall connetion done. –刷新准备工作完成

####|2017-01-16 17:43:20.476|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:20,476 INFO >>> TransAPIHandler refreshView [PROVINCE\_DEFAULT\_GROUP] Before execute. –-通过连接刷新物化视图PROVINCE\_DEFAULT\_GROUP

####|2017-01-16 17:43:25.149|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:25,149 INFO >>> TransAPIHandler refreshView [PROVINCE\_DEFAULT\_GROUP] Execute done. –刷新物化视图完成

####|2017-01-16 17:43:25.150|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:25,150 INFO >>> TransAPIHandler refreshView [PROVINCE\_DEFAULT\_GROUP] Commit done. –提交完成

####|2017-01-16 17:43:25.150|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:25,150 INFO >>> TransAPIHandler refreshView [PROVINCE\_DEFAULT\_GROUP] success, cost 4676 milliseconds. –该物化视图刷新耗时

####|2017-01-16 17:43:25.152|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:25,152 INFO >>> TransAPIHandler refreshView [PROVINCE\_DEFAULT\_GROUP] Before return true. ---该物化视图刷新结果返回true 后续是其他物化视图刷新不再重复描述

####|2017-01-16 17:43:25.152|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:25,152 INFO >>> TransAPIHandler refreshView [ROUTING\_GROUP] Before get connetion.

####|2017-01-16 17:43:25.153|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:25,153 INFO >>> TransAPIHandler refreshView [ROUTING\_GROUP] Get connetion done.

####|2017-01-16 17:43:25.154|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:25,154 INFO >>> TransAPIHandler refreshView [ROUTING\_GROUP] PrepareCall connetion done.

####|2017-01-16 17:43:25.154|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:25,154 INFO >>> TransAPIHandler refreshView [ROUTING\_GROUP] Before execute.

####|2017-01-16 17:43:26.979|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:26,979 INFO >>> TransAPIHandler refreshView [ROUTING\_GROUP] Execute done.

####|2017-01-16 17:43:26.980|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:26,980 INFO >>> TransAPIHandler refreshView [ROUTING\_GROUP] Commit done.

####|2017-01-16 17:43:26.980|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:26,980 INFO >>> TransAPIHandler refreshView [ROUTING\_GROUP] success, cost 1828 milliseconds.

####|2017-01-16 17:43:26.982|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:26,982 INFO >>> TransAPIHandler refreshView [ROUTING\_GROUP] Before return true.

####|2017-01-16 17:43:26.982|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:26,982 INFO >>> TransAPIHandler refreshView [VHOST] Before get connetion.

####|2017-01-16 17:43:26.984|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:26,984 INFO >>> TransAPIHandler refreshView [VHOST] Get connetion done.

####|2017-01-16 17:43:26.984|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:26,984 INFO >>> TransAPIHandler refreshView [VHOST] PrepareCall connetion done.

####|2017-01-16 17:43:26.984|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:26,984 INFO >>> TransAPIHandler refreshView [VHOST] Before execute.

####|2017-01-16 17:43:29.553|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:29,553 INFO >>> TransAPIHandler refreshView [VHOST] Execute done.

####|2017-01-16 17:43:29.554|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:29,554 INFO >>> TransAPIHandler refreshView [VHOST] Commit done.

####|2017-01-16 17:43:29.554|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:29,554 INFO >>> TransAPIHandler refreshView [VHOST] success, cost 2572 milliseconds.

####|2017-01-16 17:43:29.557|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:29,557 INFO >>> TransAPIHandler refreshView [VHOST] Before return true.

####|2017-01-16 17:43:29.557|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:29,557 INFO >>> TransAPIHandler refreshView [GROUP\_VHOST] Before get connetion.

####|2017-01-16 17:43:29.558|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:29,558 INFO >>> TransAPIHandler refreshView [GROUP\_VHOST] Get connetion done.

####|2017-01-16 17:43:29.558|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:29,558 INFO >>> TransAPIHandler refreshView [GROUP\_VHOST] PrepareCall connetion done.

####|2017-01-16 17:43:29.559|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:29,559 INFO >>> TransAPIHandler refreshView [GROUP\_VHOST] Before execute.

####|2017-01-16 17:43:31.382|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:31,382 INFO >>> TransAPIHandler refreshView [GROUP\_VHOST] Execute done.

####|2017-01-16 17:43:31.383|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:31,383 INFO >>> TransAPIHandler refreshView [GROUP\_VHOST] Commit done.

####|2017-01-16 17:43:31.383|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:31,383 INFO >>> TransAPIHandler refreshView [GROUP\_VHOST] success, cost 1826 milliseconds.

####|2017-01-16 17:43:31.384|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:31,384 INFO >>> TransAPIHandler refreshView [GROUP\_VHOST] Before return true.

####|2017-01-16 17:43:31.385|INFO|stream.out|\_ThreadID=1610;\_ThreadName=httpWorkerThread-1101-73;|2017-01-16 17:43:31,385 INFO >>> TransAPIHandler updateVersion: 更新t\_t\_dataset\_version表

---------------------此次FLASHTABLE操作刷新完毕

---内存刷新日志：重点注意应用获取到多少条表数据。

####|2017-01-16 18:43:58.865|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:58,865 INFO >>> TransAPI got request:FLASHTABLES

####|2017-01-16 18:43:58.866|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:58,866 INFO >>> TransAPI request action original type : memory

####|2017-01-16 18:43:58.866|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:58,866 INFO >>> TransAPI request action version : 20170116184333360

####|2017-01-16 18:43:58.877|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:58,877 INFO >>> TransAPI request action cutoffday : null

####|2017-01-16 18:43:58.877|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:58,877 INFO >>> TransAPI request action dataType : null

####|2017-01-16 18:43:58.878|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:58,878 INFO >>> TransAPI request action real type : memory

####|2017-01-16 18:43:58.878|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:58,878 INFO >>> TransAPI processing FLASHTABLES request.

####|2017-01-16 18:43:58.878|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:58,878 INFO >>> TransAPIHandler doFlashtables now.

####|2017-01-16 18:43:58.878|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:58,878 INFO >>> TransAPIHandler doFlashtables transApiActionType is memory.

####|2017-01-16 18:43:59.257|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:43:59,257 INFO >>> 739 Activity was loaded from db...

####|2017-01-16 18:44:01.065|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,065 INFO >>> 1188 ActivityAction was loaded from db...

####|2017-01-16 18:44:01.107|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,107 INFO >>> 182 Party was loaded from db...

####|2017-01-16 18:44:01.126|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,126 INFO >>> 0 PartyRateLimit was loaded from db...

####|2017-01-16 18:44:01.455|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,455 INFO >>> 1 PartyDetectList was loaded from db...

####|2017-01-16 18:44:01.465|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,465 INFO >>> 305 PartyRouting was loaded from db...

####|2017-01-16 18:44:01.478|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,478 INFO >>> 2 RecordPolicy was loaded from db...

####|2017-01-16 18:44:01.497|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,496 INFO >>> 105 ProvinceDefaultGroup was loaded from db...

####|2017-01-16 18:44:01.518|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,518 INFO >>> 27 ProvinceBipGroup was loaded from db...

####|2017-01-16 18:44:01.708|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,708 INFO >>> 400 RoutingGroup was loaded from db...

####|2017-01-16 18:44:01.724|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,724 INFO >>> 2 RoutingType was loaded from db...

####|2017-01-16 18:44:01.781|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,781 INFO >>> 647 Bips was loaded from db... --- 从数据库BIP表中获取到bip条数，其他类比理解。如果数据少了，请检查配置文件：ehcache.xml （$ECIP\_HOME/T/runtime/conf/domain\*/ ehcache.xml）,是否限制了load条数。

####|2017-01-16 18:44:01.837|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,837 INFO >>> 813 Bip Activity was loaded from db...

####|2017-01-16 18:44:01.853|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,853 INFO >>> 39 Multicast Mapping was loaded from db...

####|2017-01-16 18:44:01.867|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,867 INFO >>> 182 PartyStatus was loaded from db...

####|2017-01-16 18:44:01.908|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,908 INFO >>> 448 Vhosts was loaded from db...

####|2017-01-16 18:44:01.916|INFO|stream.out|\_ThreadID=1603;\_ThreadName=httpWorkerThread-1101-49;|2017-01-16 18:44:01,916 INFO >>> TransAPIHandler doFlashtables reloadDBConfigCache DONE. DONE. ---内存刷新完成

##### 刷新表失败

1、请根据要刷新的表，登陆对应的数据库地址，在每个需要刷新物化视图的数据库用户都手工执行刷表语句。

数据库地址和用户密码请参考表格“zkagent&db info.xlsx”。

sqlplus 用户/密码。

EXEC DBMS\_MVIEW.REFRESH('ACTIVITY','C');

EXEC DBMS\_MVIEW.REFRESH('ACTIVITY\_ACTION','C');

EXEC DBMS\_MVIEW.REFRESH('ACTIVITY\_TYPE','C');

EXEC DBMS\_MVIEW.REFRESH('BIP','C');

EXEC DBMS\_MVIEW.REFRESH('BIP\_ACTIVITY','C');

EXEC DBMS\_MVIEW.REFRESH('BIP\_TYPE','C');

EXEC DBMS\_MVIEW.REFRESH('DOMAIN','C');

EXEC DBMS\_MVIEW.REFRESH('GROUP\_VHOST','C');

EXEC DBMS\_MVIEW.REFRESH('MULTICAST\_MAPPING','C');

EXEC DBMS\_MVIEW.REFRESH('PARTY','C');

EXEC DBMS\_MVIEW.REFRESH('PARTY\_BIP','C');

EXEC DBMS\_MVIEW.REFRESH('PARTY\_RATE\_LIMIT','C');

EXEC DBMS\_MVIEW.REFRESH('PARTY\_ROUTING','C');

EXEC DBMS\_MVIEW.REFRESH('PROVINCE\_BIP\_GROUP','C');

EXEC DBMS\_MVIEW.REFRESH('PROVINCE\_DEFAULT\_GROUP','C');

EXEC DBMS\_MVIEW.REFRESH('RECORD\_POLICY','C');

EXEC DBMS\_MVIEW.REFRESH('ROUTING\_GROUP','C');

EXEC DBMS\_MVIEW.REFRESH('ROUTING\_TYPE','C');

EXEC DBMS\_MVIEW.REFRESH('SYS\_PARA','C');

EXEC DBMS\_MVIEW.REFRESH('VHOST','C');

EXEC DBMS\_MVIEW.REFRESH('MSISDN\_AREA\_MAP','C');

2、某些集群使用多数据库，比如SNRZkubl,SNkubl,KB9970等。如果这类集群或者实体应用请对集群使用到的所有数据库和数据库用户下表刷新。**其中11,13,141数据库所有数据库用户必须手工刷新。**

其中：SNkubl 使用11和13数据库的sn998和sn998\_s用户。

KB9970使用南方基地184和168数据库的sn997用户。

SNRZkubl使用的11,13,141三个数据库，11，13数据库用户为sn998,sn998\_s。141数据库使用用户：sn998，sn998\_s,sn371,sn371\_s,sn731,sn731\_s。

##### 1.2.5.4 内存刷新失败

一般不会报错，如果失败，针对失败node进行重刷操作。如果问题还没解决，请检查数据库和应用的状态以及他们之间的网络是否正常。

## 1.3生产上线检查

1、检查数据库配置正确性，在刷新全国数据前进行，此操作由当晚负责上线的人员完成。

以BIP2B248-T2101111，BIP2B247-T2001113为例，检查语句如下：

select \* from mc.t\_t\_domain where domain\_id='1258';

select \* from mc.t\_t\_party where party\_id ='12582500';

select \* from mc.t\_，t\_party\_bip where bip\_id in ('BIP2B247','BIP2B248');

select \* from mc.t\_t\_party\_routing where activity\_id in ('T2101111','T2001113');

select \* from mc.t\_t\_activity where activity\_id in ('T2101111','T2001113');

select \* from mc.t\_t\_activity\_action where activity\_id in ('T2101111','T2001113');

select \* from mc.t\_t\_bip where bip\_id in ('BIP2B247','BIP2B248');

select \* from mc.t\_t\_bip\_activity where bip\_id in ('BIP2B247','BIP2B248')

检查的重点：看插入数据库的时间，时间为当前日期；对照规范看party\_bip配置的发起落地是否正确，对照规范看party\_routing的落地配置是否正确，如果有加密业务，看一下activity\_action表的配置是否正确。

1. 检查xsd是否正确，此操作在刷新完数据库与xsd以及内存以后进行，由当晚上线负责人员安排人员完成，如果时间充裕，建议安排人员使用testtool工具测试一下报文，可以验证我们是否配置完全正确。具体测试方法请参考testtool的使用一节
2. 如果有大范围的上线或者路由调整等，刷新完数据库内存后，执行以下语句，在结果中着重看0106，0108，0113，0118，0119等中心的问题，针对性的及时解决

*--检查T模块交易是否异常*

select \* from tlog@dbs\_link\_100n\_sn100 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_2004s\_sn200 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog@dbs\_link\_2004s\_sn200 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_210n\_sn210 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_220n\_sn220 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_230f\_sn230 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_240f\_sn240 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn250.tlog@dbs\_link\_250m\_sn998 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_250n\_sn250 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_270n\_sn270 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_280n\_sn280 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_290n\_sn290 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_3115s\_sn311 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_351n\_sn351 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_371f\_sn371 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_431n\_sn431 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_451n\_sn451 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_471n\_sn471 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_531f\_sn531 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn371.tlog@dbs\_link\_998n\_141 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_571n\_sn571 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_591f\_sn591 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_731n\_sn731 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_771n\_sn771 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_791n\_sn791 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_851n\_sn851 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_871n\_sn871 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_891n\_sn891 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_891n\_sn997 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_8917s\_sn997 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_898n\_sn898 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_931f\_sn931 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_951n\_sn951 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_971n\_sn971 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_991f\_sn991 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

*--中心数据库（从11数据库往外连接）*

select \* from sn200.tlog@dbs\_link\_998n\_136 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog@dbs\_link\_998n\_136 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn852.tlog where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn250.tlog where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog@dbs\_link\_998n\_12 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_991f\_sn991 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn371.tlog@dbs\_link\_998n\_141 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog@dbs\_link\_998n\_141 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

--检查S交易是否正常

select \* from tlog@dbs\_link\_100n\_sn100 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_2004s\_sn200 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog@dbs\_link\_2004s\_sn200 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_210n\_sn210 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_220n\_sn220 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_230f\_sn230 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_240f\_sn240 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn250.tlog@dbs\_link\_250m\_sn998 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_250n\_sn250 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_270n\_sn270 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_280n\_sn280 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_290n\_sn290 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_3115s\_sn311 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_351n\_sn351 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_371f\_sn371 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_431n\_sn431 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_451n\_sn451 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_471n\_sn471 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_531f\_sn531 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn371.tlog@dbs\_link\_998n\_141 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_571n\_sn571 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_591f\_sn591 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_731n\_sn731 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_771n\_sn771 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_791n\_sn791 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_851n\_sn851 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_871n\_sn871 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_891n\_sn891 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_891n\_sn997 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_8917s\_sn997 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_898n\_sn898 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_931f\_sn931 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_951n\_sn951 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_971n\_sn971 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_991f\_sn991 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

*--中心数据库（从11数据库往外连接）*

select \* from sn200.tlog@dbs\_link\_998n\_136 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog@dbs\_link\_998n\_136 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn852.tlog where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn250.tlog where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog@dbs\_link\_998n\_12 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from tlog@dbs\_link\_991f\_sn991 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn371.tlog@dbs\_link\_998n\_141 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

select \* from sn998.tlog@dbs\_link\_998n\_141 where cutoffday='20170118' and response\_code not in ('0000','2998') order by oparty\_time desc;

# 上线常见问题

## 2.1 刷新物化视图失败

全国同时刷新数据库时，经常会有失败。其中SNkubl，SNRZkubl，KBMIGU，KB9970等尤为频繁，建议在自己的secureCRT事先配置好这些数据库机器。最好的情况是全国的数据库机器都有配置。

目前集群和实体应用混合使用，但是集群无法对多个数据库和以及一个数据库多个用户的情况物化视图刷新实现完美支持（FLASHTABLE）。因此，

**如果该集群所在的省份只有一个数据库和一个用户（只是针对一个模块）**

那么只要实体应用刷新成功，集群刷新失败可以不需要关心。目前的配置集群的node下并没有物化视图操作url。只有实体机应用刷新物化视图。所以失败的node都表示需要处理。

**如果该集群所在的省份存在多个数据库或者多个用户（只是针对一个模块）**

请手工对这些数据库下所有用户进行物化视图刷新（根据自己需要操作的表）。其中包括11,13,141这三个信息港数据库以及南方基地168和184数据库。统计如下：

其中：SNkubl 使用11和13数据库的sn998和sn998\_s用户。

KB9970使用南方基地184和168数据库的sn997用户。

SNRZkubl使用的11,13,141三个数据库，11，13数据库用户为sn998,sn998\_s。141数据库使用用户：sn998，sn998\_s,sn371,sn371\_s,sn731,sn731\_s。

文档“数据库表盒内存刷新”部分“注意事项”物化视图刷新失败时也有提醒说明。

## 2.2 刷新内存失败

1、一般不会报错，如果失败，针对失败node进行重刷操作。

2、查看T或S应用节点Server日志是否收到请求，如果是，请调到步骤4.如果否，请继续步骤3

3、如果T或S没收到请求，请按照文档“数据库与内存刷新-注意事项-界面反馈结果错误处理方法”进行检查和处理。

4、确定检查数据库和应用的状态以及他们之间的网络是否正常。

应用server日志如果出现如下几种报错。一般会短信告警，这类情况请在群里反馈并申请重启应用的操作。（如果联系不上，请自行重启实例，并记录）

JVM out of memory

cat not create native thread

java heap space

如果重启不了，请登录BES 修改JVM配置，将JVM改小尝试再次重启。

## 2.3 0126 bip not match activity

请检查对应应用连接的数据库用户下的物化视图bip，activity，bip\_activity表是否配置了上发报文中的bip和activity。特别需要关注集群使用的数据库为多个的情况，对于SNkubl，SNRZkubl，KB9970这类特殊集群更加需要关注。最好记住这几个集群使用了哪些数据库。

## 2.4 OSN/HSN init error(0112 ，0301?)

如果是个别应用出现这种情况，一般是内存刷新没成功，请重新对该应用刷新内存。

## 2.5 switchpin null(0112,0301)

这种错误表示应用无法对报文中的密文进行正确加解密。

如果是新上的业务，请检查activity\_action表是否配置错误。（非加密业务配置成加密业务或者加密节点路径不对）。

如果是确定配置没有问题，请检查该应用调用的加密机服务是否启动，否则启动加密机。

如果加密机服务存在，那么加密字符串，例如：123 让发起方解密或发起方加密相同字符串让中心解密或者对比。

注意：有的机构存在多个加密机的情况，如果手工加解密没问题，请咨询下他们的加密机是否存在多个。

## 2.6 上线配置不全

1、先将已经能上线的业务先上线。

2、获取规范并到测试环境获取业务配置信息。（业务配置信息不包含路由信息表，例如：vhost,group\_vhost,routing\_group,province\_default\_group,province\_bip\_group）

如果只是上业务，请不要动路由信息相关的表。

1. 替换party\_routing中落地地址为生产地址（测试环境配置的是生产地址）
2. Party表中LOCAL\_REQ\_URL一般为LOCALHOST。
3. Xsd直接从测试环境获取，并根据文档“xsd上传”部分上传，刷新内存即可。

以上操作一般秀娟已经完成，上线人员操作只需要刷新物化视图和内存即可。但是请熟悉上线的资源获取来源和步骤，对突发问题的解决有很大的帮助。

## 2.7应答报文发起方解析不了，转换xml失败。

这个问题主要出现在同duns osn直接给落地方转发，落地方应答报文不规范。目前来看，主要是存在于客服之间。由于osn并不对落地报文进行非常详尽的兼容性处理，所以osn在收到报文并转发给发起方时，对于某些兼容性差的发起方可能会出现解析不了的问题。这种情况需要根据三方确定解决方法。一般party表中落地方LOCAL\_REQ\_URL

如果设置URl地址，那么这个问题可以解决。但是这种设置涉及到与这个落地方同duns的所有到这个落地方的业务都要走LOCAL\_REQ\_URl这个地址。影响比较大。之前遇到的是二线客服给一级客服发报文，但是二线客服解析不了一线客服的应答。如果修改一线客服的party表配置，那么所有二线客服都将调到新的 LOCAL\_REQ\_URl这个地址然后落地到一级客服，对于现有的二线客服可能会有影响，所以最终是一级客服要求二线客服修改。这里重点是存在这种现象，需要有个理解。

## 2.8上线刷新party\_bip表导致未注册问题。

处理方法：

由于签到签退会修改MC.t\_t\_party\_bip表，上线新业务时也会配置这张表。如果这时候有机构已经签退。则会将签退业务刷新到省数据库物化视图。但是按照要求省里的物化视图party\_bip表必须是签到状态相违背。所以针对这种情况，请上线时，必须检查mc.t\_t\_party\_bip除了已经集团要求签退的业务是否存在签退业务或者机构。

集团要求的签退机构

MBOX2000,MBLG2000

签退业务如下：

BIP5A025 BOSS8910 1 0 0 1

BIP5A025 BOSS8910 1 1 0 1

因此上线时，请保证省份数据库party\_bip表除了上面集团要求的签退业务和平台，其他所有机构和业务都必须是签到状态。如果当时mc.t\_t\_party\_bip表存在签退的机构和业务（非要求签退的机构和业务），请刷新party\_bip表和内存时，在群里通知省份不要在你操作的时候做签到签退操作。并且在在party\_bip表刷新内存之后，将省里的物化视图恢复到签到状态。Mc.t\_t\_party\_bip保证是是实时的签到签退状态。

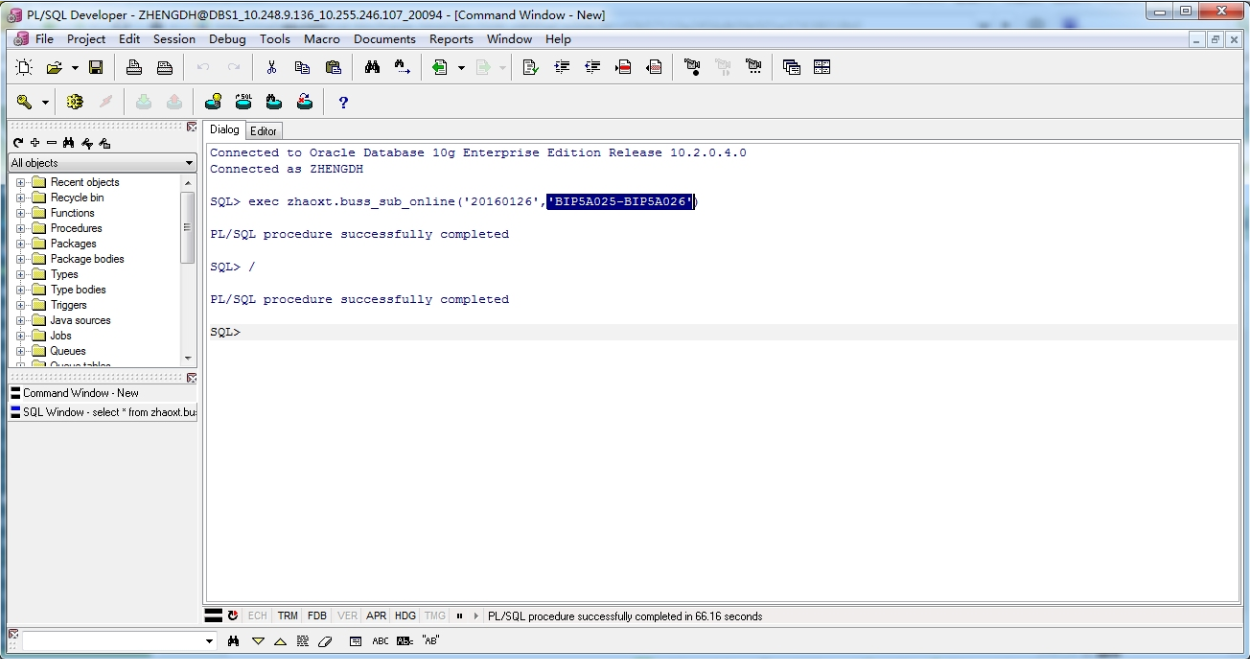
## 2.9 新上线xsd报0106，0108

如果本次上线有上新的xsd，上线过程中报出0106，0108等检验失败的情况，主要先通过校验工具校验下报文，校验成功就检查xsd版本、有没有上传以及内存是否刷新成功。

# 日常生产运维

## 生产上线后数据统计（主要用于出报告）（小龙）

1. 登录136数据库确认 mc.t\_bussiness\_golive\_repo，rt表中的DBLINK正确性，通过select t.\*,rowid from mc.t\_bussiness\_golive\_report t where is\_valid='1'语句 查询的结果是所有的数据库DBLINK。
2. 表zhaoxt.buss\_sub\_online\_tmp存储执行存储过程数据，如果觉得表的数据多余可以清除。通过Truncate table zhaoxt.buss\_sub\_online\_tmp语句清除以前统计的冗余数据。
3. plsql 打开command 窗口。执行exec zhaoxt.buss\_sub\_online('日切日','BIP1-BIP2-BIP3....');
4. 例如：exec zhaoxt.buss\_sub\_online('20160126','BIP5A025-BIP5A026');



1. 上线完成后一般需要执行两次，一次执行的是上线当天的日切日，一次是上线第二天日切日。两者数据集叠加才是需要统计的数据集合。
2. 执行group by语句查询zhaoxt.buss\_sub\_online\_tmp表统计接口测试结果。

## Party置成维护状态（小龙）

该处以设置VGOP为维护状态和恢复维护状态为例。其他机构同理。

**设置VGOP维护状态**

1、登陆10.1.128.113；

2、进入目录： cd /opt/ecip/BMC/hadoop-nv/zookeeper/bin/；

3、进入zookeeper控制台： ./zkCli.sh -server localhost:1195；

4、进入控制台后，将VGOP9980置为维护状态：set /conf/BMC/check/maintParties VGOP9980；

5、退出ctrl+c退出控制台；

6、等待手机告警VGOP party status fail，否则检查之前步骤。

**恢复VGOP机构状态**

1、登陆10.1.128.113；

2、进入目录： cd /opt/ecip/BMC/hadoop-nv/zookeeper/bin/；

3、进入zookeeper控制台： ./zkCli.sh -server localhost:1195；

4、进入控制台后，将VGOP9980置为维护状态：set /conf/BMC/check/maintParties  1；

5、退出ctrl+c退出控制台；

6、VGOP party 状态告警消失，VGOP恢复正常。否则检查之前步骤。

## 日切及其处理方法（宁波）

每晚23点06分会准时日切，成功短信如下：

Node OP9990,OP5710,OP2000,FTPSERVER,run cutoffday and cutoff backup has Exception..

如果短信中有SN或KB开头的如SN9980U，则这个节点日切异常，需要登录内网或者外网进行检查处理。

### 2.3.1外网日切

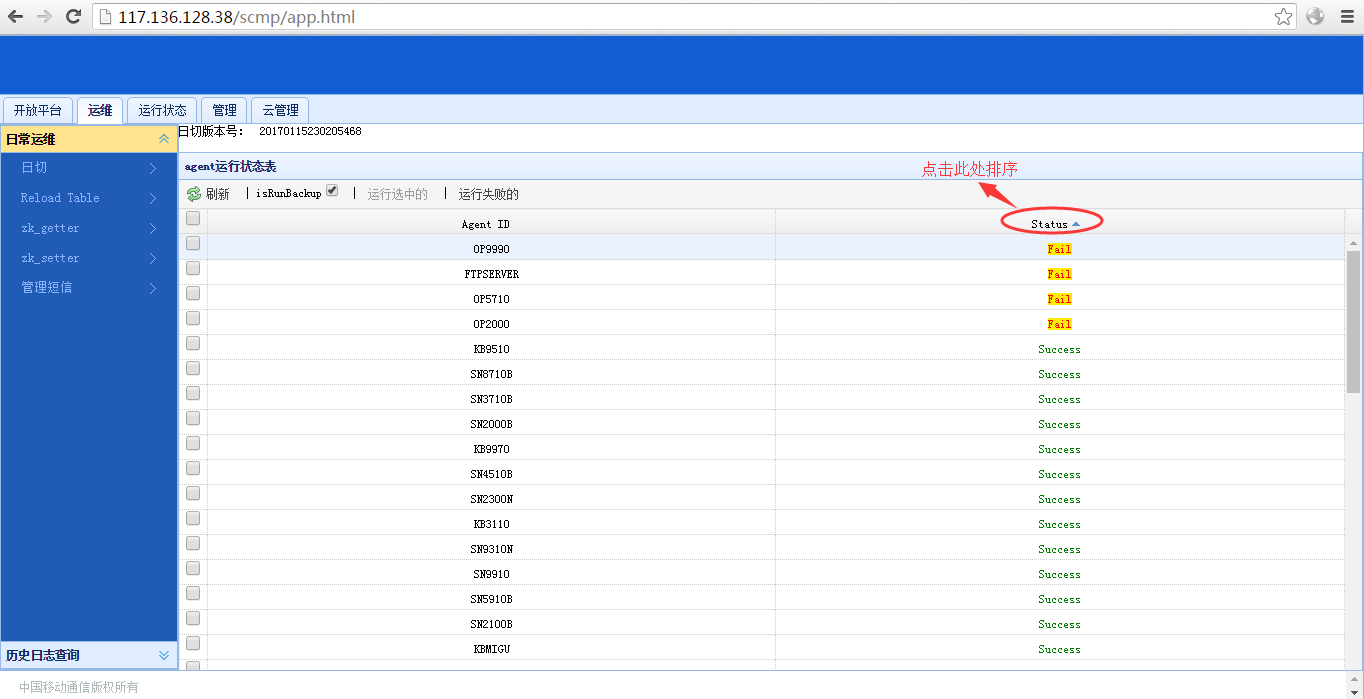
1、登陆网状网开放平台管理系统

打开Google Chrome，登入http://117.136.128.38/scmp/login.html。（如果手机号登录不了请找赵仁东将你号码赋权；注意只能用谷歌浏览器；短信验证码是和告警短信一个号码发送)

2、通过外网执行日切

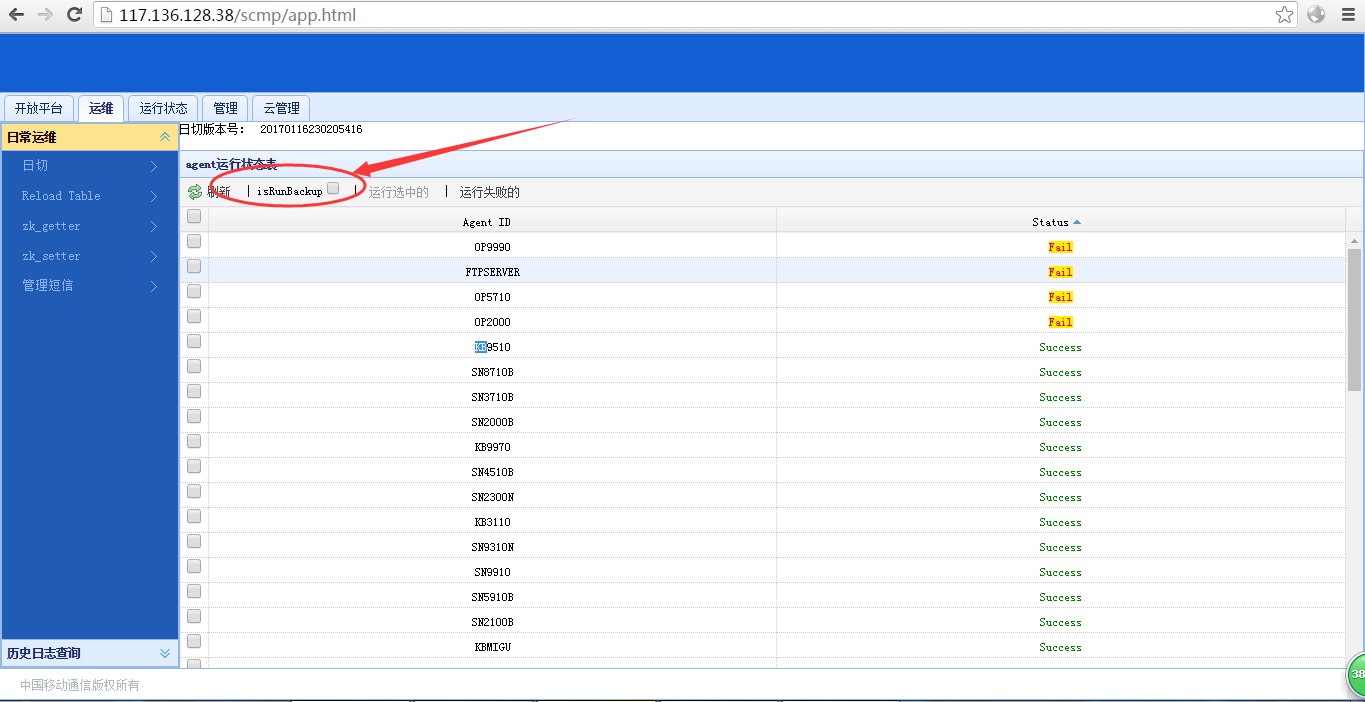
* 情况1：非k8s集群日切（既SN+省DUNS号，如SN9980U）

成功登陆后点击运维-日切，在日切失败Agent前打上“√”然后点击运行选中的，收到弹框提醒：“你好，因为当前操作属于异步操作，运行该程序后页面不会自动刷新，确认运行3分钟后请刷新当前页面。若操作结果存在异常或其他错误，请查看具体日志（下一步），选择“YES”,刷新几次后Agent对应的Status为“Success”则日切成功，如果长时间为“Fail”或“Running”则进行2.3.2；



* 情况2：k8s集群日切（以KB开头的省为k8s集群，如KB9510）

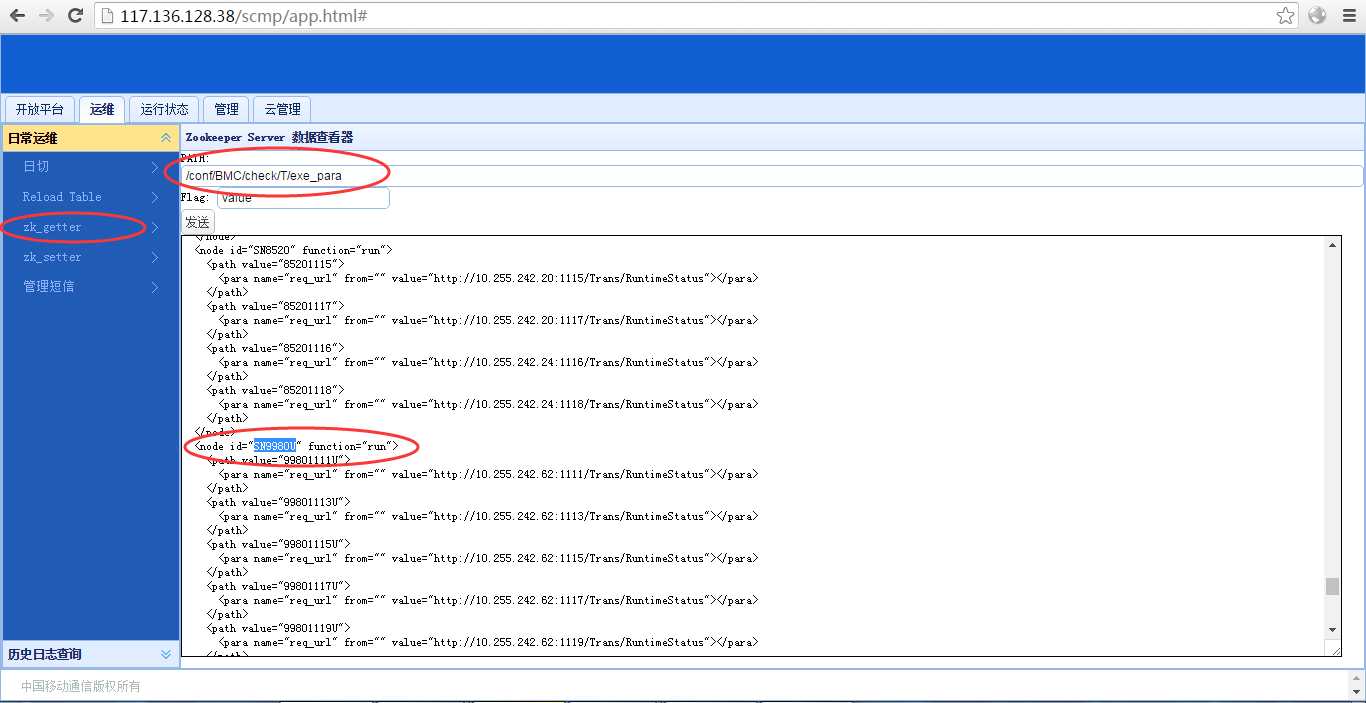
成功登陆后点击运维-日切，在日切失败Agent前打上“√”，然后将isRunBackup的对勾去掉，点击运行选中的，收到弹框提醒：“你好，因为当前操作属于异步操作，运行该程序后页面不会自动刷新，确认运行3分钟后请刷新当前页面。若操作结果存在异常或其他错误，请查看具体日志（下一步），选择“YES”,刷新几次后Agent对应的Status为“Success”则日切成功，如果长时间为“Fail”或“Running”则进行2.3.2；



### 2.3.2 定位失败的agent

1、 找到日切失败的agent对应的服务器ip

在http://117.136.128.38/scmp/login.html页面中点击zk\_getter,然后将PATH的值改为“/conf/BMC/check/T/exe\_para”点击发送，然后按Ctrl+f搜索日切失败的省如SN9980U，找到Agent对应的机器为10.255.242.62



2、找到agent对应的zk的位置

使用4a跳转到10.255.242.62机器，执行命令

env | grep TW （找到SN9980U对应zkagent的ip）

TWSECU\_SERVER=10.255.242.62

登陆到10.255.242.62

cd $ECIP\_HOME

cd zkAgent

cat run.sh （确定此为SN9980U的zk）（下面为run.sh中的内容）

/\*export BASH\_SHELL\_PATH=/bin/bash

nohup ./startZkAgentService.sh SN9980U 10.1.128.113:1195,10.248.9.135:1195,10.255.242.11:1195,10.255.242.22:1195,10.255.242.24:1195 C&\*/

### 2.3.3重新启动zk

找到agent后，查看一下agent的日志，如果能够定位原因，按照原因处理，如果无法定位，则可以重启一下agent实验一下。

重启ZkAgent步骤如下

#ps -aux | grep zk (找到zk的进程，注意linux和hp-ux的使用不用的ps参数，hp-ux使用ps -exf )

ecip 51172 2.8 3.2 11993892 1051652 ? Sl 2016 1410:10 /opt/jdk1.6.0\_45/bin/java -cp .:./lib/dom4j.jar:./lib/jaxen.jar:./lib/bzu.jar:./lib/jline.jar:./lib/log4j.jar:./lib/netty.jar:./lib/slf4j.jar:./lib/slf4j\_log.jar:./lib/zk.jar com.hp.ecip.zk.utils.ZkAgentService SN9980U 10.1.128.113:1195,10.248.9.135:1195,10.255.242.11:1195,10.255.242.22:1195,10.255.242.24:1195

可以看到SN9980U这个zk的进程

#kill -9 51172 （杀掉SN9980U的进程）

#cd $ECIP\_HOME

#cd zkAgent

#./run.sh (启动zk)

#cd logs

#tail -n 100 zk\_agent.log (查看日志)

查看日志显示zk正常启动无报错，继续在外网日切执行2.3.1中的步骤2，显示“Success”后成功

**异常情况：**

日志中显示zk无法正常启动且报错出现连接，可能为zk连接数量过多导致的，修改run.sh

中连接server的数量可以解决，如

可以将5的server中的2个去掉，只剩3个server

export BASH\_SHELL\_PATH=/bin/bash

nohup ./startZkAgentService.sh SN9980U 10.255.242.11:1195,10.255.242.22:1195,10.255.242.24:1195 C&

修改成功后再次启动，查看zkagent日志正常启动无报错，继续在外网日切执行中2.3.1的步骤2，显示“Success”后成功；

### 2.3.4检查节点的日切情况

查看该节点对应服务器所有Instance OSN日志，执行命令：

#cd $ECIP\_HOME

#cd /T/oltplogs/osn

#ll -rt （查看最新的osn日2.3.1志文件名，找到有2个，分别检查日切情况）

-rw-r----- 1 ecip appadm 19108230 Jan 16 15:35 OSN\_OLTPLOG\_99801119

-rw-r----- 1 ecip appadm 6748662 Jan 16 17:06 OSN\_OLTPLOG\_99801117

#tail OSN\_OLTPLOG\_99801117 (注意红色部分为第二天才是日切成功)

20170116170645545| INFO : Thread-1441>>>20170116|BBSS|POCP|BIP4B776|T4101047|SUF2017011616545723124269|9980111720170116165457379521384|Notice Reload Dispatcher 1\_3 Got XmlMessage

20170116170645545| INFO :

#tail OSN\_OLTPLOG\_99801117

20170116153533460| INFO : Thread-1442>>>20170116|BBSS|POCP|BIP4B776|T4101047|SUF2017011615250823056734|9980111920170116152508171481072|Notice Reload Dispatcher 1\_4 Got XmlMessage

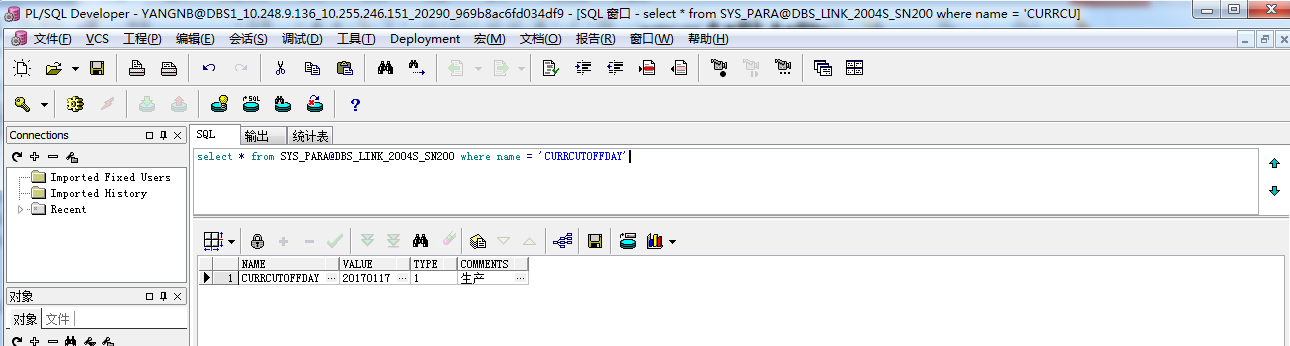
### 2.3.5生产网日切

原理：

日切是T从内存中取出，每天的晚上23:06分MC会准时修改数据库t\_t\_sys\_para表，并通知各省刷新sys\_para物化视图和内存。目前的失败主要是因为物化视图刷新失败。

#### 2.3.5.1检查物化视图是否刷新失败

从4a登陆136或11数据库，使用对应的dblink查询失败agent对应数据库sys\_para表的日切信息是否正常（第二天）如SN200的如图（select \* from SYS\_PARA@DBS\_LINK\_2004S\_SN200 where name = 'CURRCUTOFFDAY'），如果此时的日期正常为第二天，跳过刷新物化视图，直接刷新agent对应的内存见2.3



注意'CURRCUTOFFDAY'的VALUE没有切换到第二天需要进行物化视图刷新2.2

#### 2.3.5.2物化视图刷新失败处理方法

1、视图刷新方法1

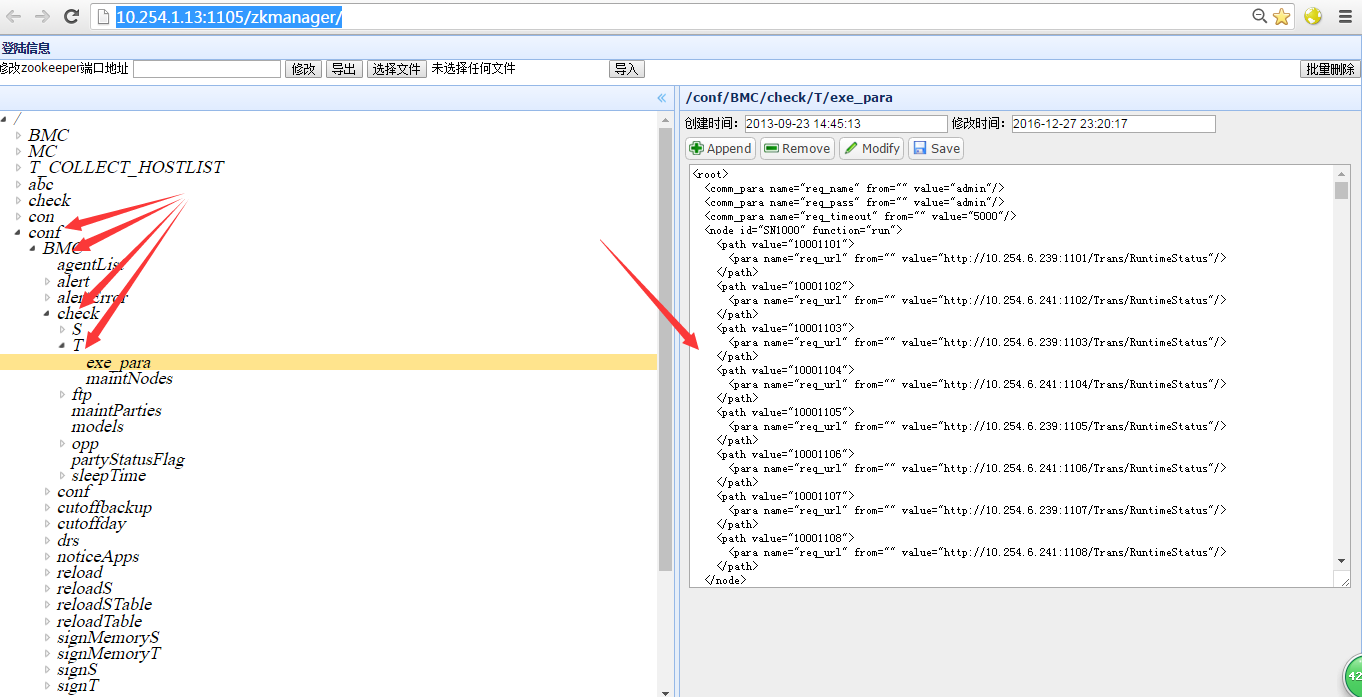
当数据库刷新失败时，刷新物化视图。方法有两种：第一种到是失败省数据库直接语句刷新。

EXEC DBMS\_MVIEW.REFRESH('SYS\_PARA','C');

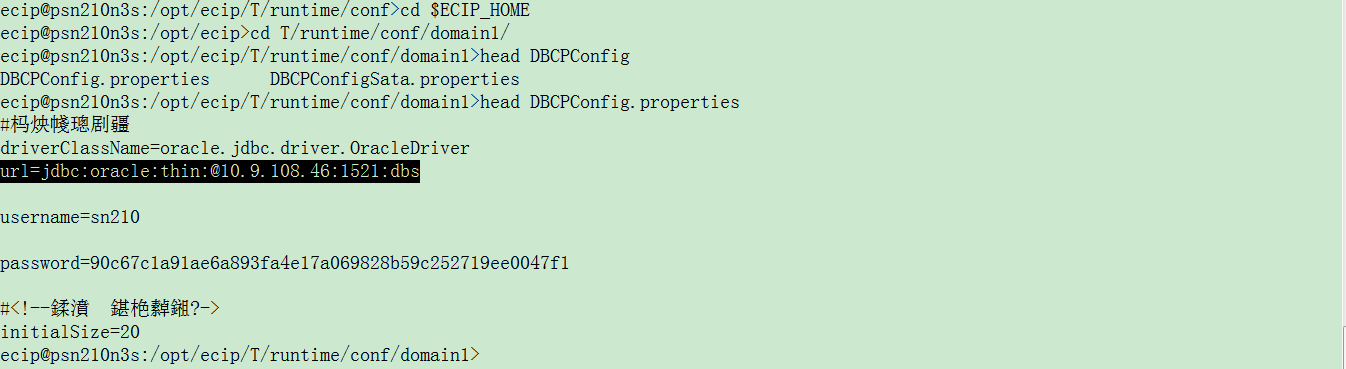
可以通过在http://10.254.1.13:1105/zkmanager/下找到失败的agent对应的服务器IP如

SN2100B对应的ip为10.9.108.66

如图：



找到应用所对应的数据库ip并登陆如图



ssh登陆省数据库10.9.108.46

成功登陆后

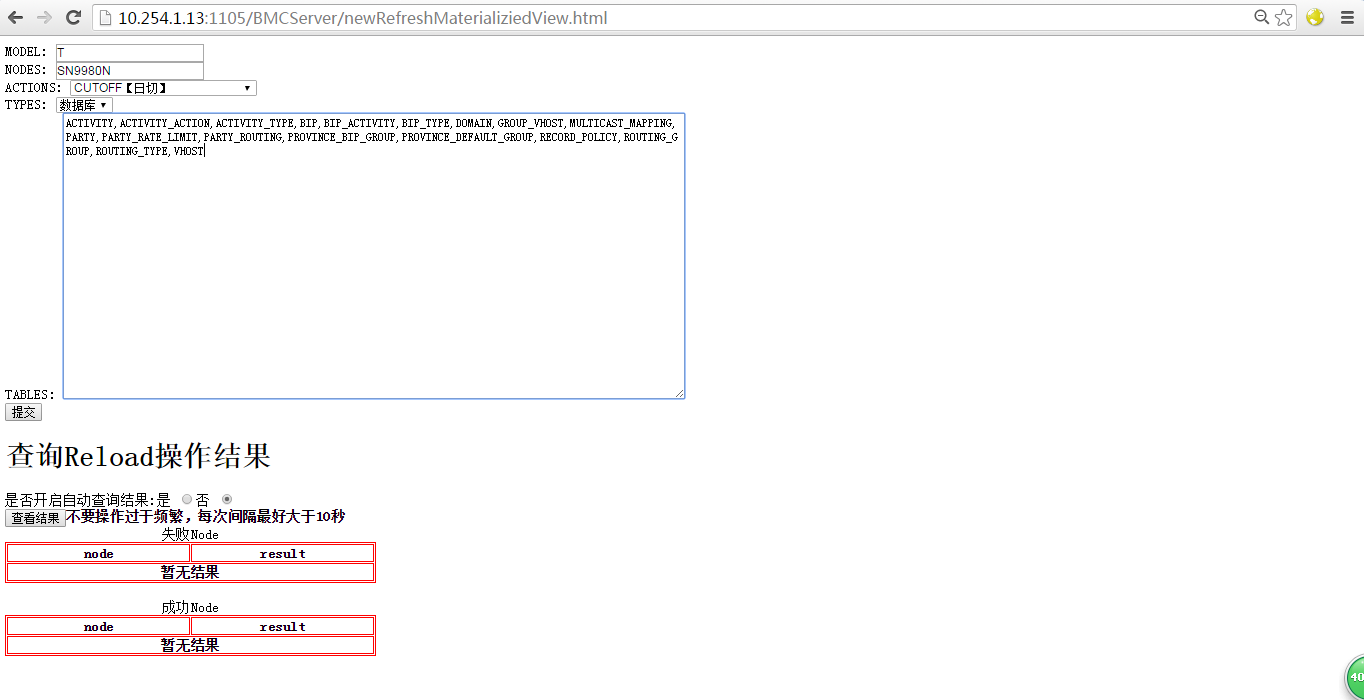
#sqlplus sn210/sn210 (以sn210用户登陆)

EXEC DBMS\_MVIEW.REFRESH('SYS\_PARA','C') （手动刷新SYS\_PARA的物化视图）

2、**视图刷新**方法2

登陆BMCServer(http://10.254.1.13:1105/BMCServer/newRefreshMaterializiedView.html)前台页面，执行type为“数据库”，action为“CUTOFF”日切的工具刷新。注意NODES为失败省所在NODE(如SN9980N)，type为数据库表示只刷数据库。

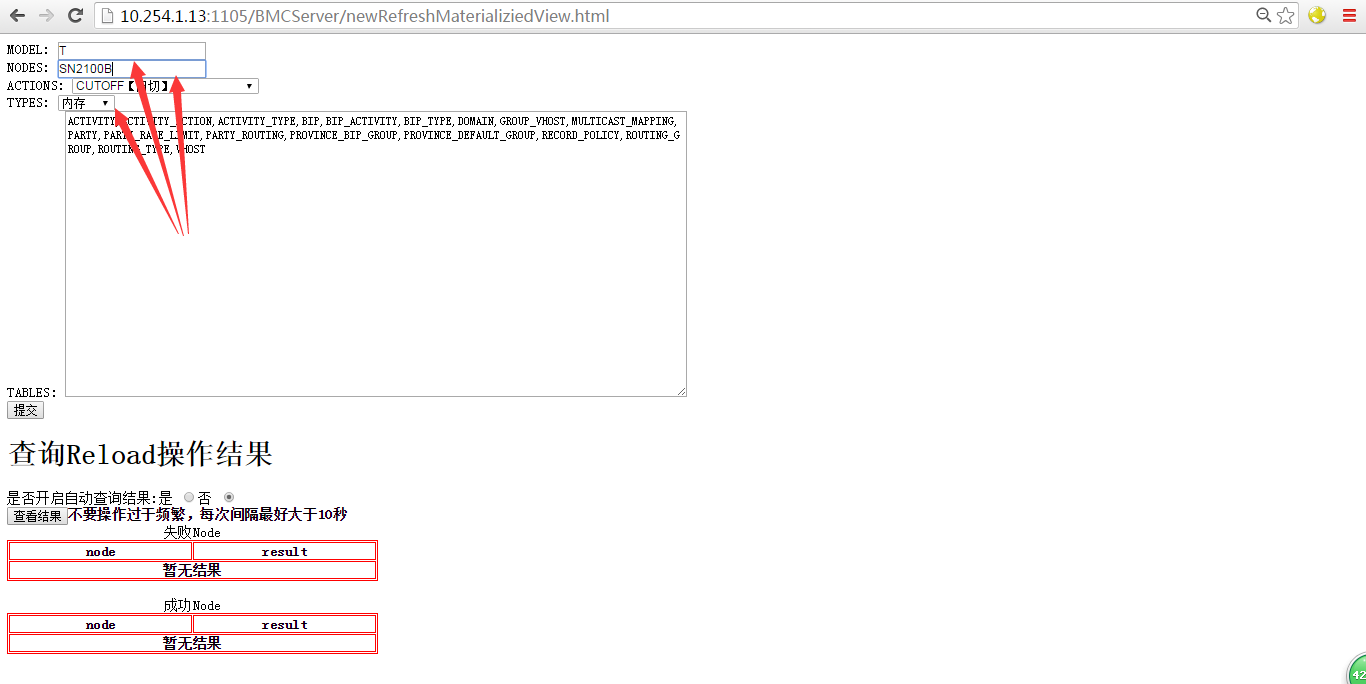
如图：



注意：如果此时返回的结果为失败，需要检查并重启zk，方法见于2.3.2和2.3.3。然后重新进行此步骤

#### 2.3.5.3刷日切内存方法

只是刷新内存失败，那么通过BMCServer（http://10.254.1.13:1105/BMCServer/newRefreshMaterializiedView.html）前台页面搞定，区别之前截图type选择为内存即可，以SN2100B为例子，如下图：



注意：如果此时返回的结果为失败，需要检查并重启zk，方法见于2.3.2和2.3.3。然后重新进行此步骤

#### 2.3.5.4检查节点的日切情况

具体方法同2.3.4

## 1.6 zookeeper重启（东海）

Zookeeper集群由于各种原因，经常进程会消失。目前是正对这个情况采用脚本自动重启。

但是如果需要手工重启zookeeper起参考如下步骤。

1. 确定zookeeper所在机器：

在zkagent安装机器，ps -exf|grep zk，查看结果，可以看到zookeeper所在机器和端口（高亮部分）。

ecip@psn2005s:/opt/cmcb/ap5/ecip/T/var/domains/domain1/nodes/na/instances/aloneIns1/logs> ps -exf|grep zk

ecip 23666 23665 0 Jun 24 ? 191:53 /opt/java6/bin/IA64N/java -cp .:./lib/dom4j.jar:./lib/jaxen.jar:./lib/bzu1.5.1.jar:./lib/jline.jar:./lib/log4j.jar:./lib/netty.jar:./lib/slf4j.jar:./lib/slf4j\_log.jar:./lib/zk.jar com.hp.ecip.zk.utils.ZkAgentService SN2000 10.254.1.13:1195,10.254.1.11:1195,10.254.1.23:1195 C

ecip 29778 25641 0 14:54:12 pts/ta 0:00 grep zk

1. 确定目录：

telnet到对应的zookeeper安装机器，ps -exf|grep zoo得到结果：

ecip@cmcbnscs:/opt/cmcb/csc/ecip>ps -exf|grep zoo

ecip 19506 1 0 Apr 30 ? 96:25 /opt/java6/bin/PA\_RISC2.0/java -Dproc\_zkapp -Xms16M -Xmx32M -XX:+HeapDumpOnOutOfMemoryError -XX:+HeapDumpOnCtrlBreak -Dtest.home=/opt/ecip/BMC/hadoop-nv/zkapp -Dhadoop.log.dir=/opt/ecip/BMC/hadoop-nv/zkapp/logs -Dhadoop.log.file=zk.log -classpath /conf:/opt/ecip/BMC/hadoop-nv/zkapp/conf:/opt/ecip/BMC/hadoop-nv/zkapp/eciptzk.jar:/opt/ecip/BMC/hadoop-nv/zkapp/lib/jline-0.9.94.jar:/opt/ecip/BMC/hadoop-nv/zkapp/lib/log4j-1.2.15.jar:/opt/ecip/BMC/hadoop-nv/zkapp/lib/netty-3.2.2.Final.jar:/opt/ecip/BMC/hadoop-nv/zkapp/lib/slf4j-api-1.6.1.jar:/opt/ecip/BMC/hadoop-nv/zkapp/lib/slf4j-log4j12-1.6.1.jar:/opt/ecip/BMC/hadoop-nv/zkapp/lib/zookeeper-3.4.3.jar:/opt/ecip/BMC/hadoop-nv/zkapp/lib/\*.zip:/opt/ecip/BMC/hadoop-nv/zkapp/lib/xmlbeans/\*.jar -Dfile.encoding=utf-8 com.hp.ecip.zk.ZkDemo cnt113

ecip 16019 1 0 May 27 ? 3794:58 /opt/java6/bin/PA\_RISC2.0/java -Dzookeeper.log.dir=. -Dzookeeper.root.logger=INFO,CONSOLE -cp /opt/ecip/BMC/hadoop-nv/zookeeper/bin/../build/classes:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../build/lib/\*.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/slf4j-log4j12-1.6.1.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/slf4j-api-1.6.1.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/netty-3.2.2.Final.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/log4j-1.2.15.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/jline-0.9.94.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../zookeeper-3.4.3.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../src/java/lib/\*.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../conf: -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.local.only=false org.apache.zookeeper.server.quorum.QuorumPeerMain /opt/ecip/BMC/hadoop-nv/zookeeper/bin/../conf/zoo.cfg

ecip 24517 24513 0 14:16:31 pts/tc 0:05 /opt/java6/bin/PA\_RISC2.0/java -Dzookeeper.log.dir=. -Dzookeeper.root.logger=INFO,CONSOLE -cp /opt/ecip/BMC/hadoop-nv/zookeeper/bin/../build/classes:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../build/lib/\*.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/slf4j-log4j12-1.6.1.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/slf4j-api-1.6.1.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/netty-3.2.2.Final.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/log4j-1.2.15.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../lib/jline-0.9.94.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../zookeeper-3.4.3.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../src/java/lib/\*.jar:/opt/ecip/BMC/hadoop-nv/zookeeper/bin/../conf: org.apache.zookeeper.ZooKeeperMain -server localhost:1195

ecip 15058 15040 0 14:56:23 pts/tf 0:00 grep zoo

以上黄色字体就是zookeeper的启动脚本所在。

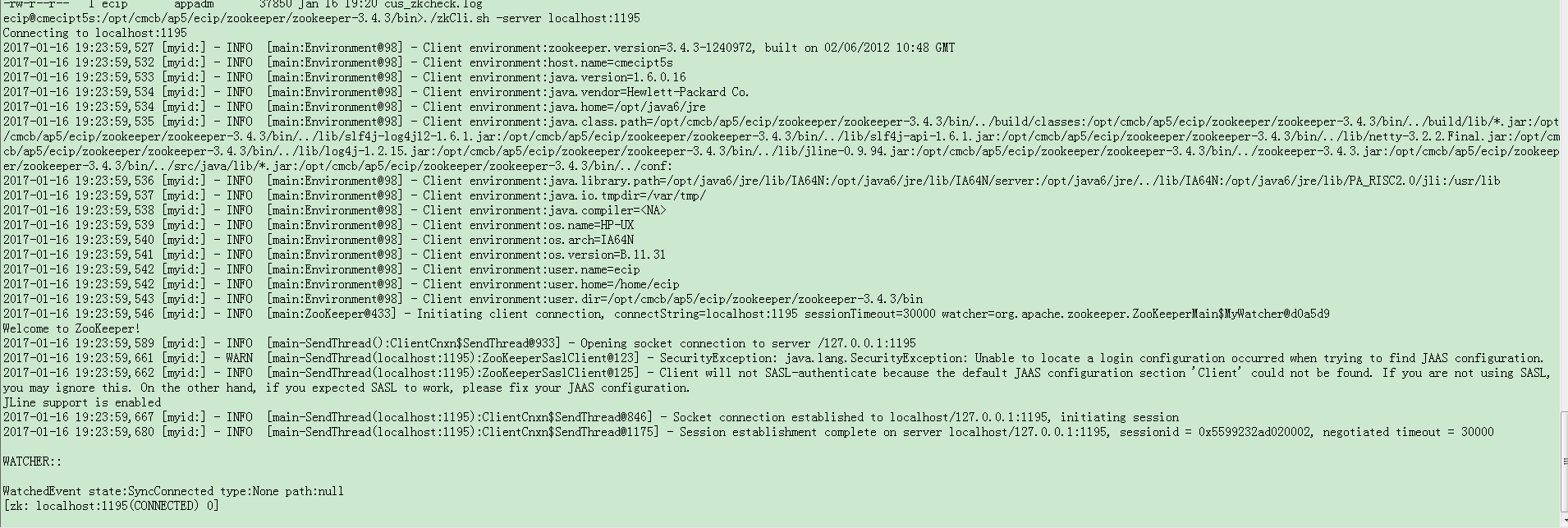
1. 命令

下面是作为一个非开发人员，经常需要用一般性的命令，请先进入到zookeeper所在目录：

zookeeper服务的启动：bin/zkServer.sh start

zookeeper服务的关闭：bin/zkServer.sh stop （可以直接kill然后通过命令启动）

登陆zookeeper控制台：bin/zkCli.sh -server ip:port



Zookeeper日志文件：bin/zookeeper.out

Zookeeper配置文件：conf/zoo.cfg

1. 验证方法

登陆zookeeper控制台：bin/zkCli.sh -server ip:port 没有任何报错即可。

某些情况下需要所有zookeeper server重启，一般不需要。

目前生产zookeeper server所在机器和路径如下：

10.255.242.24: ecip/Um\_yH%1lD

[ecip@cmecipt6s:/opt/cmcb/ap6/ecip/zookeeper/zookeeper-3.4.3/bin](mailto:ecip@cmecipt6s:/opt/cmcb/ap6/ecip/zookeeper/zookeeper-3.4.3/bin)>

10.255.242.22： ecip/Um\_yH%1lD

[ecip@cmecipt5s:/opt/cmcb/ap5/ecip/zookeeper/zookeeper-3.4.3/bin](mailto:ecip@cmecipt5s:/opt/cmcb/ap5/ecip/zookeeper/zookeeper-3.4.3/bin)>

10.255.242.11 ecip/Um\_yH%1lD

ecip@cmdbs1s:/opt/cmcb/safiles/ecip/zookeeper/bin

10.1.128.113 ecip/ecip

ecip: /opt/ecip/BMC/hadoop-nv/zookeeper/bin

10.255.242.135 ecip/Ecip.123!

ecip: /hadoop/ecip/T\_zookeeper/zookeeper/bin

## 1.7 重启agent节点（东海）

一般情况下，zkagent出现问题时，重启是通用解决方法。

Zkagent 地址请参考表格“zkagent&db info.xlsx”。（加密机和zkagent是必须待一起的，除非zkagent不需要参与tws功能）

1、ecip用户登陆对应机器

2、根据表格找到zkagent路径(一般$ECIP\_HOME或/opt/ecip 或者ap0外挂盘路径)。可以尝试使用命令：

find / -name zkagent (unix,linux),

locate –bi zkagent (linux)

查找

3、ps -exf|grep zk，查看是否是即所找的zkagent名字

4、 kill 掉进程号

5、 ecip@cmcscbks:/opt/cmcb/bks/ecip/zkAgent>Sending output to nohup.out

脚本运行完成

6、进入logs目录,tail -f zk\_agent.log,查看日志是否有报错。如下是SN2000N正常启动的日志：注意红色字段为机器bash路径，请确保正确。

0116 19:14:19.252[com.hp.ecip.zk.utils.ZkAgentService(138)] INFO||SN2000N================ZooKeeper Agent Was Start================

0116 19:14:19.256[com.hp.ecip.zk.utils.ZkAgentService(139)] INFO||ENV SHELL PATH [BASH\_SHELL\_PATH]=>>null DEFAULT=>>/usr/local/bin/bash

0116 19:14:19.744[com.hp.ecip.zk.client.CheckThread(106)] INFO||SN2000N =====model [T] ===== Check Thread init over =====

0116 19:14:19.745[com.hp.ecip.zk.client.CheckThread(247)] INFO||SN2000N =====model>>T<<===== Check Thread Start =====

0116 19:14:20.259[com.hp.ecip.zk.client.AlertNodes(32)] INFO||SN2000N=====model [T] ========== Alert Nodes Start ===================

0116 19:14:20.828[com.hp.ecip.zk.client.CheckThread(106)] INFO||SN2000N =====model [S] ===== Check Thread init over =====

0116 19:14:20.830[com.hp.ecip.zk.client.CheckThread(247)] INFO||SN2000N =====model>>S<<===== Check Thread Start =====

0116 19:14:21.338[com.hp.ecip.zk.client.AlertNodes(32)] INFO||SN2000N=====model [S] ========== Alert Nodes Start ===================

0116 19:14:21.393[com.hp.ecip.zk.client.AlertNodes(42)] WARN||can not fand exe para by Model[S] <==>SN2000N please check you zk para data...

0116 19:14:21.861[com.hp.ecip.zk.client.CheckOppThread(75)] INFO||SN2000N =====model [opp] ===== Check Thread init over =====

0116 19:14:21.863[com.hp.ecip.zk.client.CheckOppThread(80)] INFO||SN2000N =====model>>opp<<===== Check Thread Start =====

0116 19:14:21.874[com.hp.ecip.zk.client.CheckFtpThread(96)] INFO||SN2000N =====model [ftp] ===== Check Thread init over =====

0116 19:14:21.875[com.hp.ecip.zk.client.CheckFtpThread(101)] INFO||SN2000N =====model>>ftp<<===== Check Thread Start =====

0116 19:14:35.838[com.hp.ecip.zk.client.CheckThread(258)] WARN||can not fand exe para byModel[S] <==> SN2000N please check you zk para data...

0116 19:14:36.867[com.hp.ecip.zk.client.CheckOppThread(91)] WARN||can not fand exe para byModel[opp] <==> SN2000N please check you zk para data...

0116 19:14:36.887[com.hp.ecip.zk.client.CheckFtpThread(115)] WARN||can not find exe para byModel[ftp] <==> SN2000N please check your zk para data...

7、同时ps -exf|grep zk查看zkagent是否启动成功。如果5分钟之后进程还在表示启动一般没有问题。

ecip@cmecipt4s:/home/ecip>cd $ECIP\_HOME

ecip@cmecipt4s:/opt/cmcb/ap4/ecip>ps -exf|grep zk

ecip 23484 23483 0 Dec 28 ? 363:19 /opt/java6/bin/IA64N/java -cp .:./lib/dom4j.jar:./lib/jaxen.jar:./lib/bzu.jar:./lib/jline.jar:./lib/log4j.jar:./lib/netty.jar:./lib/slf4j.jar:./lib/slf4j\_log.jar:./lib/zk.jar com.hp.ecip.zk.utils.ZkAgentService SN2000N 10.1.128.113:1195,10.1.128.111:1195,10.248.9.136:1195,10.255.242.11:1195 C

ecip 18876 18809 0 19:13:09 pts/0 0:00 grep zk

ecip@cmecipt4s:/opt/cmcb/ap4/ecip>

## 1.8短信告警处理方法（东海）

### 1.8.1 instance不可用

短信：[T/S] Warning check instance :37101140,37101138,37101136, error.. 探测到应用状态不正常

处理方法：

如果只是短暂的告警某个或某些应用状态不对：

表示当时业务量很大或者落地方应答比较慢，经过隔离一段时间，落地方服务恢复，T/S应用也自动恢复。这种情况不需要处理，但是这可能意味着落地方服务或者中心实例需要扩容。

如果长时间收到该告警：

请根据zookeeper中配置的功能check/T|S/exe\_para下找到对应的告警实例，查看该实例的server日志。这里主要可能原因是在大业务量情况下，应答过慢或者超时。应答超时，可以在server日志看到报错日志，主要有read time out，connection time out等报错信息。但是如果是应答过慢，这种问题则在server日志看不到error日志。请在短信告警期间查看省boss oltp日志（一般为hsn日志）中业务量比较大的落地业务（一般为UMMP（S模块UACP），UPSS等）。此类问题一般为省boss出现，如果中心出现一般为系统资源问题。

### 1.8.2 插库失败

短信：[T]37101114Instance find some XMLMessages was not inserted Datebase! \ [T] 37101114Instance Database insert OK now! 插入数据库失败

处理方法：

此类问题主要跟T/S插入数据库问题相关。前面短信表示应用插入数据库失败。后面短信表示插入数据库成功。如果在接下来的一分多钟内接收到第二条短信插入数据库成功。那么可能是偶然问题（可能是网络，可能是业务量大问题），不需要紧急处理。如果是频繁的报插入数据库失败的短信，需要紧急处理。

一般情况是表空间满，或者分区问题。此类问题请按照如下步骤解决：

旧有数据库系统组会有定期的清理，不会出现问题，如果是某天业务量特别特别大。可能导致来不起清理就已经告警。

1. 登录应用对应数据库机器（参考“zkagent&db info.xlsx”数据库信息），使用对应的用户名/密码登录数据库，

set linesize 300;

select TABLE\_NAME,PARTITION\_NAME,TABLESPACE\_NAME from user\_tab\_partitions t where t.TABLE\_NAME='TMSG' order by 1,2;

select TABLE\_NAME,PARTITION\_NAME,TABLESPACE\_NAME from user\_tab\_partitions t where t.TABLE\_NAME=’TLOG’ order by 1,2;

查询出来的partition按照保留三个月的原则，三个月前的partition可以删除（如果已经告警那么按照最早的日期删除一天到3天），先恢复生产。删除语句：

alter table tmsg drop partition PT\_TMSG\_20120920(删除语句，注意操作，大写部分为查出来的分区名);

alter table tlog drop partition PT\_TLOG\_20120920(删除语句，注意操作，大写部分为查出来的分区名);

注意：如果一个省的TMSG空间保留不了一个月的数据，TLOG空间保留不了3个月数据，请及时提出来告知新涛，需要扩容。

1. 在原来的用户下，使用下面的语句查看目前表空间使用率到达什么情况，一般情况下，清除后剩余使用率在50以上即可。

select a.tablespace\_name "tablespace name",

round(a.bytes\_alloc/1024/1024,2) "size(m)",

--round((a.bytes\_alloc-nvl(b.bytes\_free,0))/1024/1024,2) "used(m)",

round(nvl(b.bytes\_free,0)/1024/1024,2) "free(m)",

round((nvl(b.bytes\_free,0)/a.bytes\_alloc)\*100,2) "free pecent(%)"

--to\_char(sysdate,'yyyy-mm-dd hh24:mi:ss') "sample time"

from (select f.tablespace\_name,

sum(f.bytes) bytes\_alloc,

sum(decode(f.autoextensible,' yes',f.maxbytes,' no',f.bytes)) maxbytes from dba\_data\_files f group by tablespace\_name) a, (select f.tablespace\_name,

sum(f.bytes) bytes\_free

from dba\_free\_space f

group by tablespace\_name) b

where a.tablespace\_name = b.tablespace\_name

order by a.tablespace\_name;

注意：如果执行这个语句没有权限，可以使用管理员账户登录看sqlplus "/ as sysdba"

如果是OSN的server日志中出现 can not get datasource connection， timeout wait for idle datasource connection等报错，表示应用和数据库之间存在供求关系不匹配。一般先通过数据库查看单位时间分钟交易量来确定osn是否足够支撑当时的上发速度，极少情况下会发生如此大的业务上发量变化，需要向发起方确认是否存在异常，如果该上发量会一直持续，则需要对osn扩容。

对于大业务量平台上发osn来说， 数据库可能成为瓶颈（手机支付，统一支付，移动商城，统一认证等），这类问题只能采用分流到多个数据库来支撑。其中

手机支付在容灾集群有osn，入库到141数据库。本地入湖南数据库。

移动商城同时存在mysql数据库，11数据库，13数据库。

统一认证：11数据库，13数据库。

统一支付：南方基地168数据库，184数据库

还有一种可能是应用跟数据库之前存在网络问题，此类问题会在server日志中，体现各种连接被拒绝，连接超时等。这种情况需要联系网络处理。请反馈到系统组。

### 1.8.3 机构状态不正常

短信： [T] Warning check parties status :MEBP5710, is fail 检查到机构状态不正常

处理方法:

网状网会根据单位时间内，落地方失败条数来判断落地方是否正常。这里的失败条数包含0133,0134等转发失败的情况。如果某个机构频繁报该问题，请找到该机构的落地hsn，查看hsn的server日志定位问题，并反馈到机构联系人。(qq，邮件，电话等方式)

### 1.8.4 落地机构处理失败交易较多

短信：20170117133254542[BOSS2900T:21/NIAP2400T:18/]PartyBizFailed in 10 minutes! 10分钟机构失败条数

处理方法：

表示过去的十分钟内，机构失败的条数。这里包含0108,0133,0134等非正常状态的交易。如果是连续告警且失败量很大，请查看server日志定位原因。如果偶然告警且量不大处理优先级不高。

这种情况一般为落地服务不稳定造成，可能是落地机构承载业务量过大导致，也可能是网络不稳定导致。如果查看日志存在http状态不对（500,503,404），则需要联系落地方解决。如果存在http 400 或者连接断断续续，则需要ping 落地地址或者多次curl落地地址确认网络是否正常。网络一般会伴随有机构状态不正常告警。

### 1.8.5 zookerper连接丢失

短信：xxxx==>>check instance run time error:KeeperErrorCode = ConnectionLoss for xxxx zookeeper连接丢失

处理方法：

这种问题出现一般表示zookeeper集群大部分进程被停掉了。请参考zookeeper重启查看zookeeper日志，记录原因。一般为can not create native thread。 当然也不排除被杀掉的可能。通常直接启动zookeeper即可。如何启动zookeeper请参考“zookeeper重启”部分详细描述。目前有脚本每隔五分钟会检查zookeeper，应该出现问题的可能性比较小。当然如果系统资源不足以启动zookeeper，则需要联系系统组处理。

### 1.8.6 签到

短信：[BOSS2200]Sign in Node OP9990,OP2000,OP5710,FTPSERVER, sign memory has Exception 签到

处理方法：

该短信表示某机构签到结果。这里表示天津签到有几个node没成功。这些都不是T S的zkagent所以不需要关心。但是如果出现这几个node之外的zkagent签到失败，则需要检查应用是否正常。由于目前签到签退都为内存操作，一般T和S的状态正常，收到请求都能正常处理。所以如果失败，请先检查zkagent日志是否给T或S转发http 请求，如果没有转发请求，请重启zkagent。然后刷新该zkagent的party\_bip内存即可。（内存签到签退只修改t\_t\_party\_bip表，省里面的物化视图必须是签到状态）

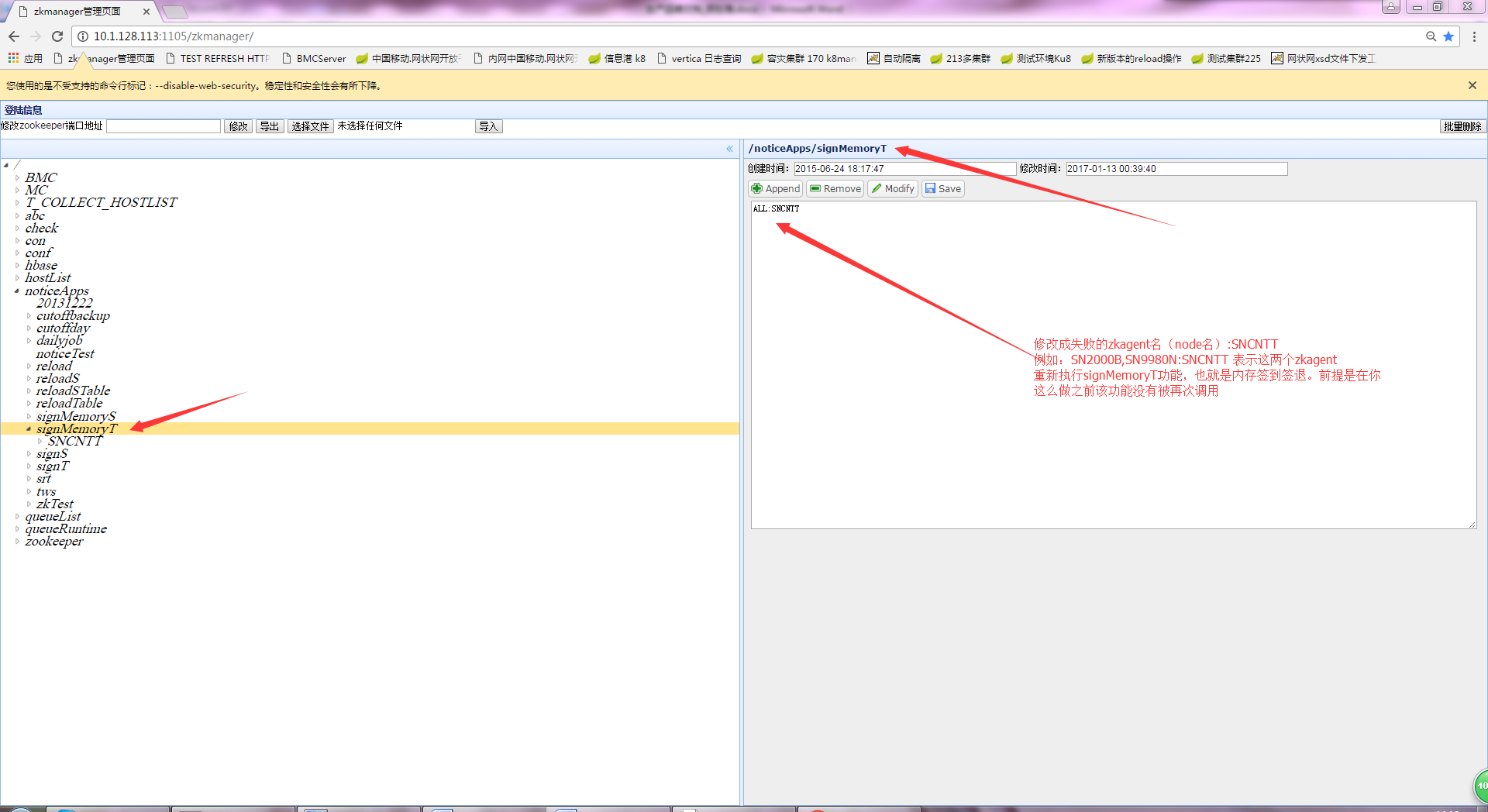
### 1.8.7 签退

短信：[IRCN9970]Sign out Node OP9990,OP2000,OP5710,FTPSERVER, sign memory has Exception

处理方法：

该短信表示某机构签退结果。这里表示IRCN9970签退有几个node没成功。这些都不是T S的zkagent所以不需要关心。但是如果出现这几个node之外的zkagent签到失败，则需要检查应用是否正常。由于目前签到签退都为内存操作，一般T和S的状态正常，收到请求都能正常处理。所以如果失败，请先检查zkagent日志是否给T或S转发http 请求，如果没有转发请求，请重启zkagent。

通过zkmanager修改内存签到签退功能的触发node值，触发失败的zkagent重复之前的操作。请注意注释部分。该操作只能重复最近的一个内存签到签退内容。



如果需要出发上一个内存签到签退功能之外的签到。请取到上次发起方发起的上发报文，

根据报文修改t\_t\_party\_bip表中对应的party satus和bip status为签到状态。让发起方或者自己模拟发起方上发签退报文再次触发全国内存签到。

### 1.8.8定时任务入库失败

短信：[T]99801101Instance insertTLOG failed,totalFailFileCount:[1],totalFailCount:[1] 每天0点入库当日失败sql情况

处理方法：

表示当日写入文件的失败sql，执行重新插入数据库的时候失败的文件数和sql条数。每个instance单独告警。如果有失败的情况，请根据实例名查看server日志，查看失败的sql的convid，通过搜索runtime/document/sql/目录下的实例名.sql\*,查看整个convid相关的所有sql。

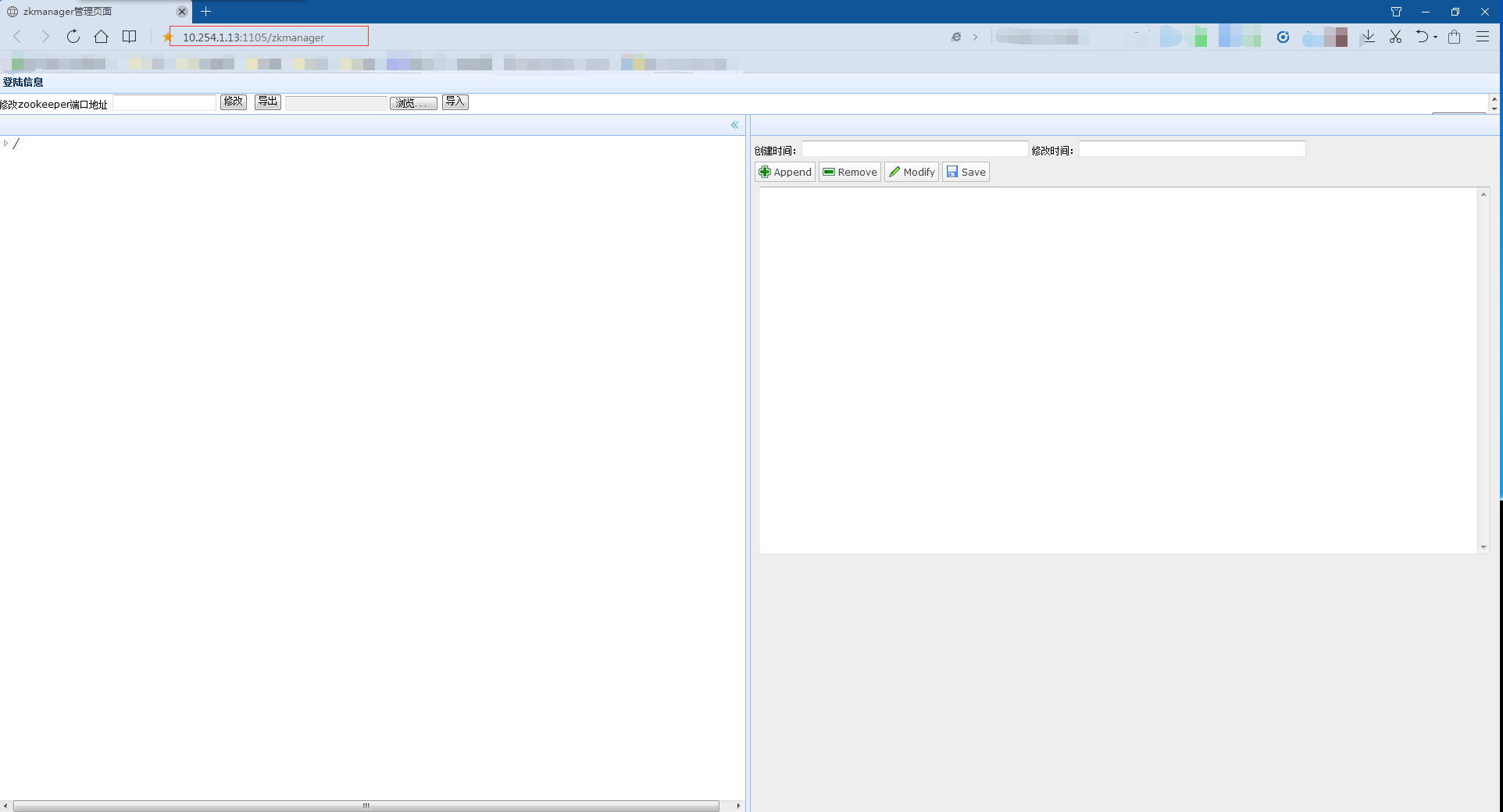
再查看数据库中该convid记录是否完整，来确定是否需要重新执行该sql。登陆到实例连接的数据库机器，sqlplus 用户/密码。执行搜索到跟该convid相关的所有sql。最后commit; 确认提交。

目前为止暂时没有发现对生产有影响的此类问题。

# 生产zookeeper配置(景阳)

## 3.1查看zookeeper配置

Zookeeper操作目前采用页面的形式，用浏览器访问地址(推荐谷歌浏览器)：<http://10.1.128.113:1105/zkmanager/>

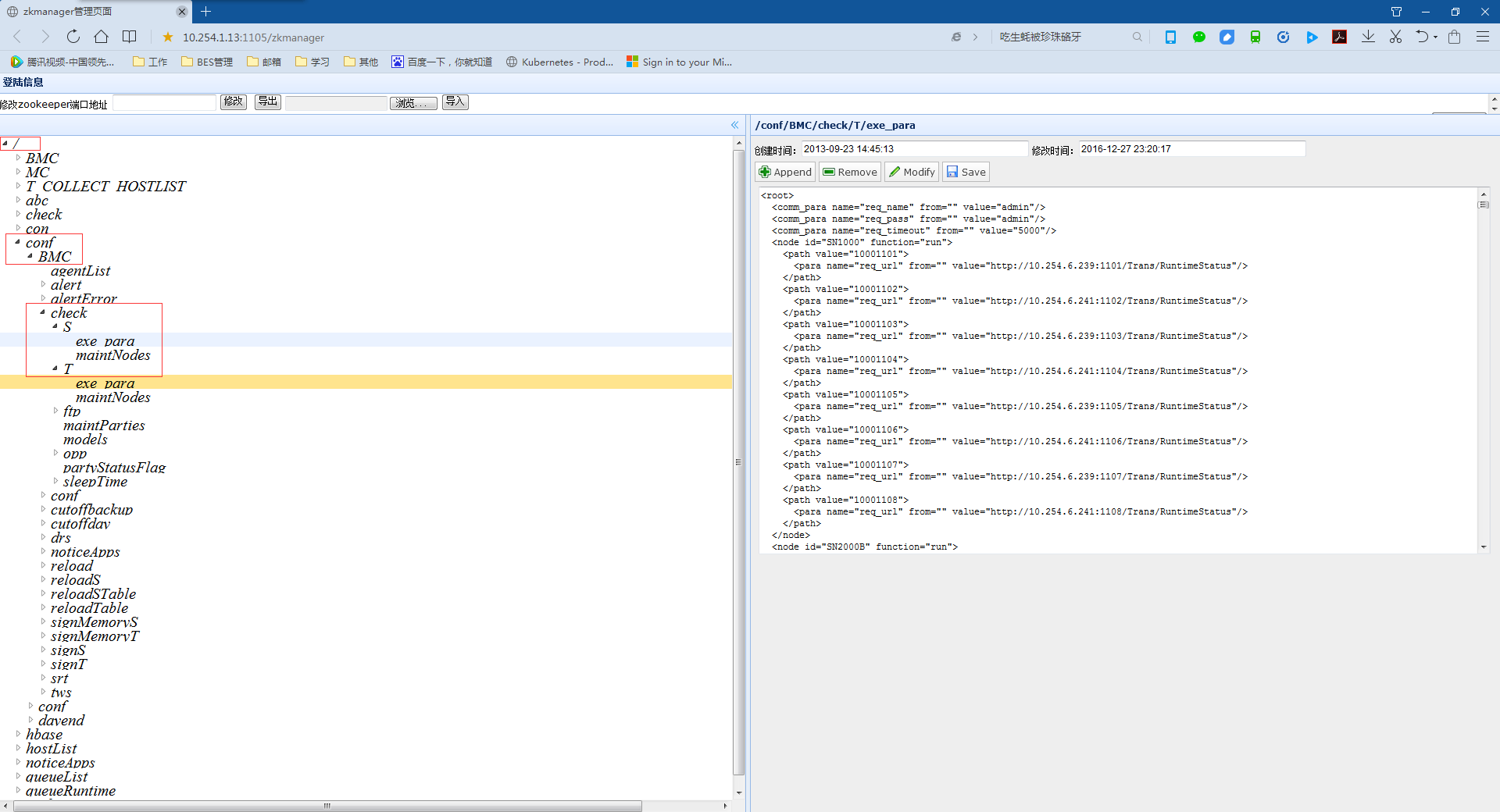


点击依次进入T的配置，路径：**/conf/BMC/check/T/exe\_para**

S的配置，路径：**/conf/BMC/check/S/exe\_para**

**\*不要进错目录，操作之前检查目录是否正确，请谨慎操作！**

**如图所示：**

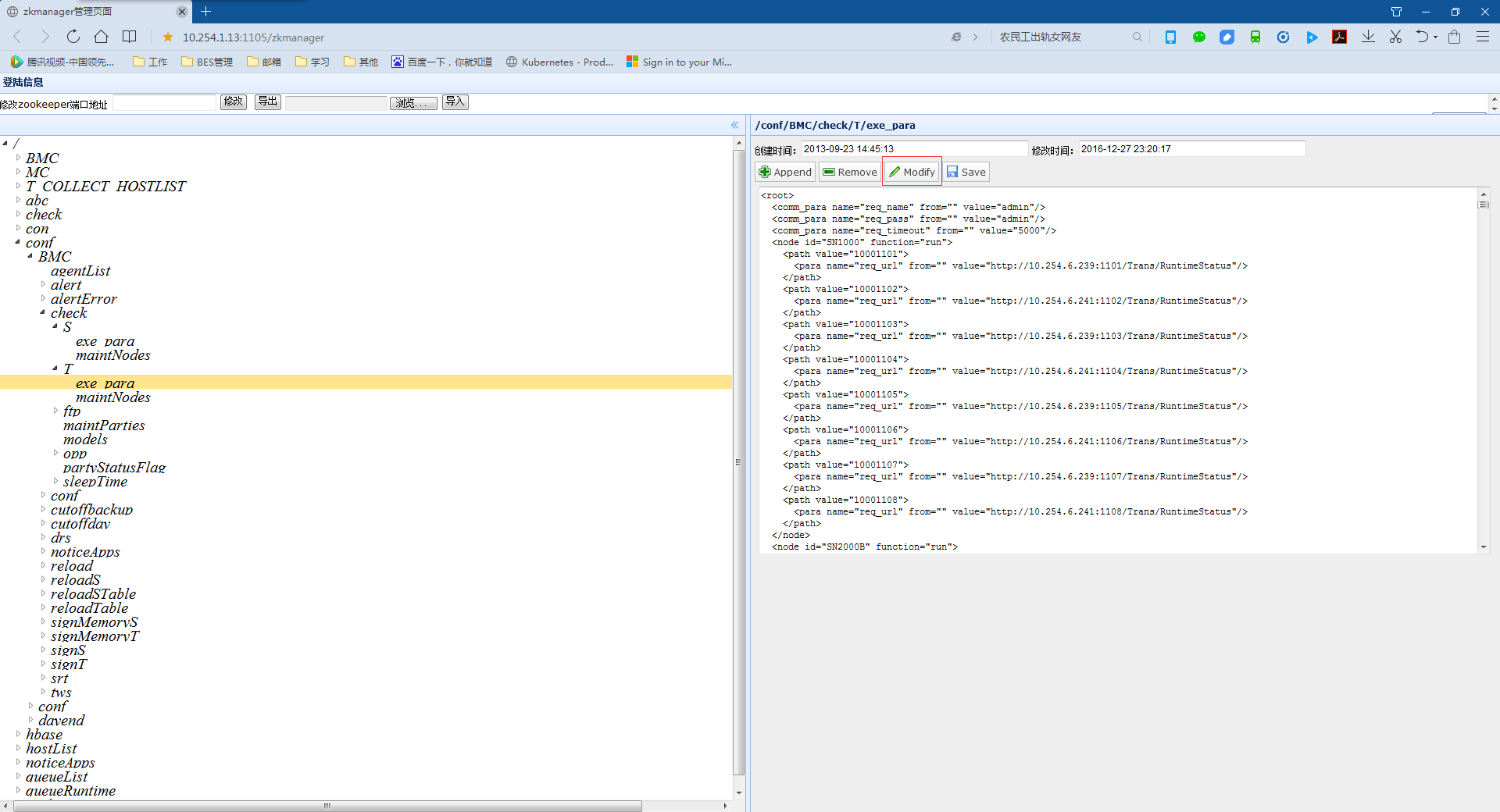


## 3.2修改zookeeper配置（请谨慎操作！）

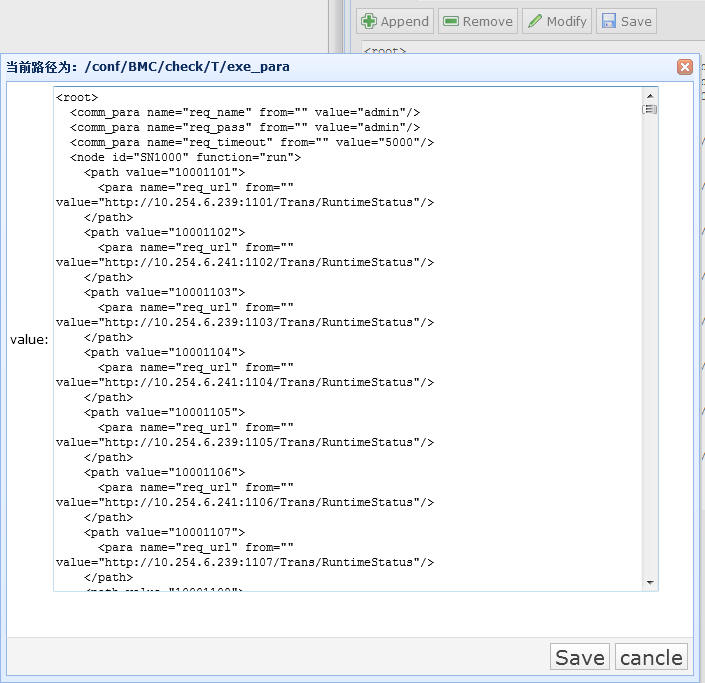
### 3.2.1修改T的zookeeper配置

1.选择路径：**/conf/BMC/check/T/exe\_para**

2.右侧选择：Modify按钮 



进入编辑界面：



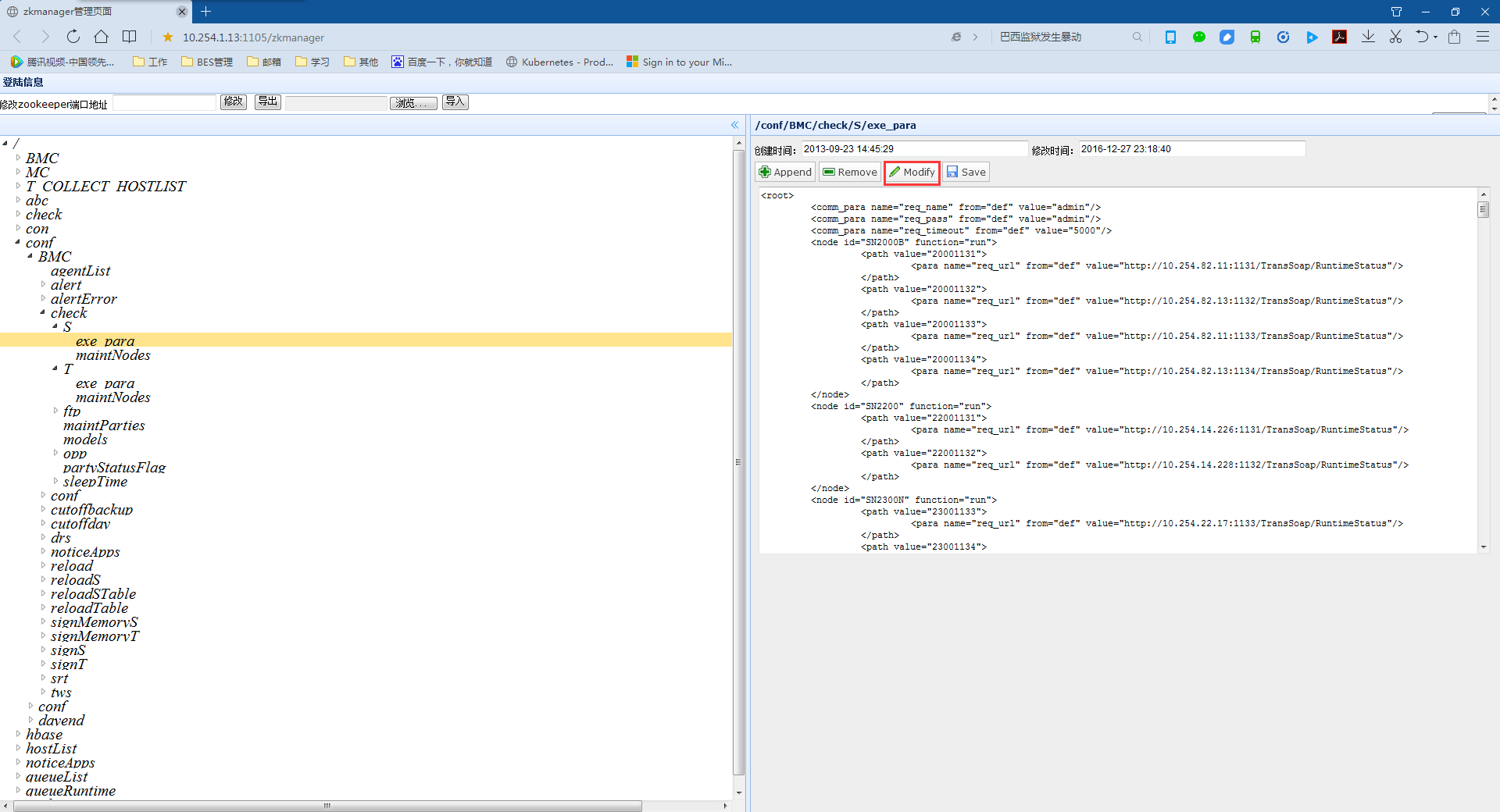
4.保存配置内容之前请校验配置的内容是否为一个正确的xml。

5.Save保存，cancle退出

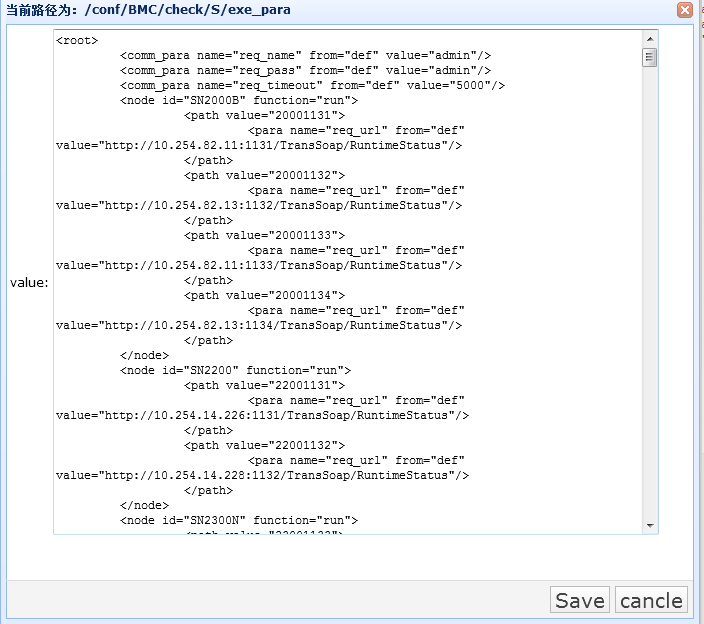
### 3.2.2修改S的zooker配置

1.选择路径：**/conf/BMC/check/S/exe\_para**

2.右侧选择：Modify按钮 



进入编辑界面：



4. 保存配置内容之前请校验配置的内容是否为一个正确的xml。

5.Save保存，cancle退出

## 3.3同理修改其他配置项

**配置操作同上，具体操作路径和功能如下：（**务必保证配置内容为xml格式。**）**

T，S内存刷新

**T：/conf/BMC/reload/exe\_para**

**S：/conf/BMC/reloadS/exe\_para**

T，S的表刷新

**T：/conf/BMC/reloadTable/exe\_para**

**S：/conf/BMC/reloadSTable/exe\_para**

T，S内存签到签退

**T：/conf/BMC/signMemoryT/exe\_para**

**S：/conf/BMC/signMemoryS/exe\_para**

旧T，S签到签退：

**T：/conf/BMC/signT/exe\_para**

**S：/conf/BMC/signS/exe\_para**

普通日切：

**/conf/BMC/cutoffday/exe\_para**

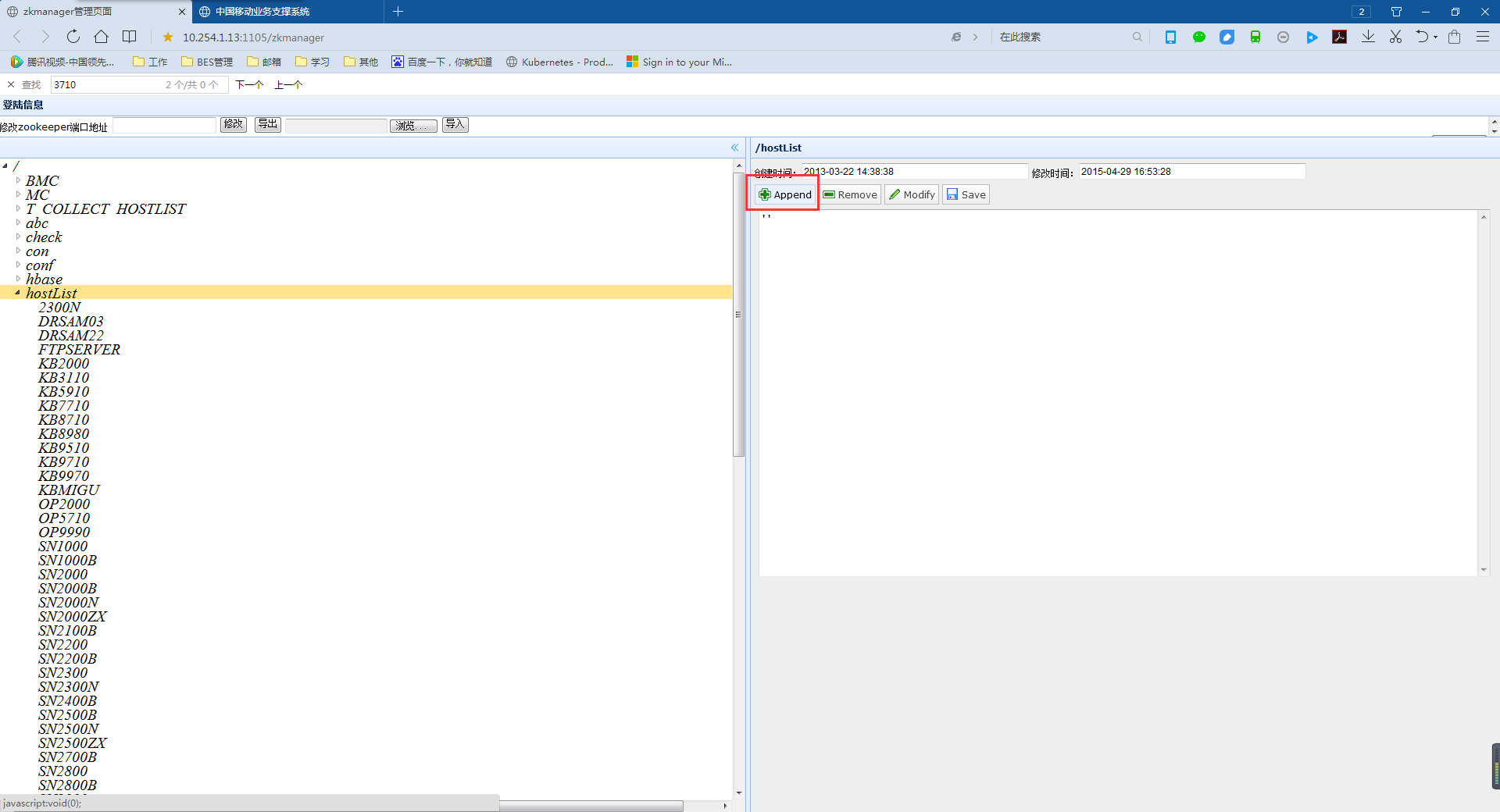
内存日切：

**/conf/BMC/cutoffbackup/exe\_para**

## 3.4加密机：

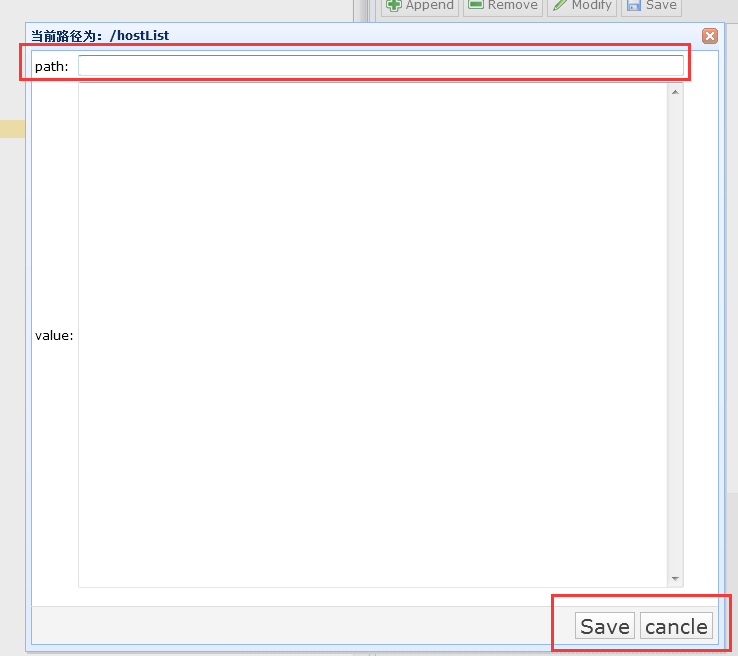
### 3.4.1加密机zokeeper配置

1. **注册node，要在/hostlist append 增加相应的node名字**

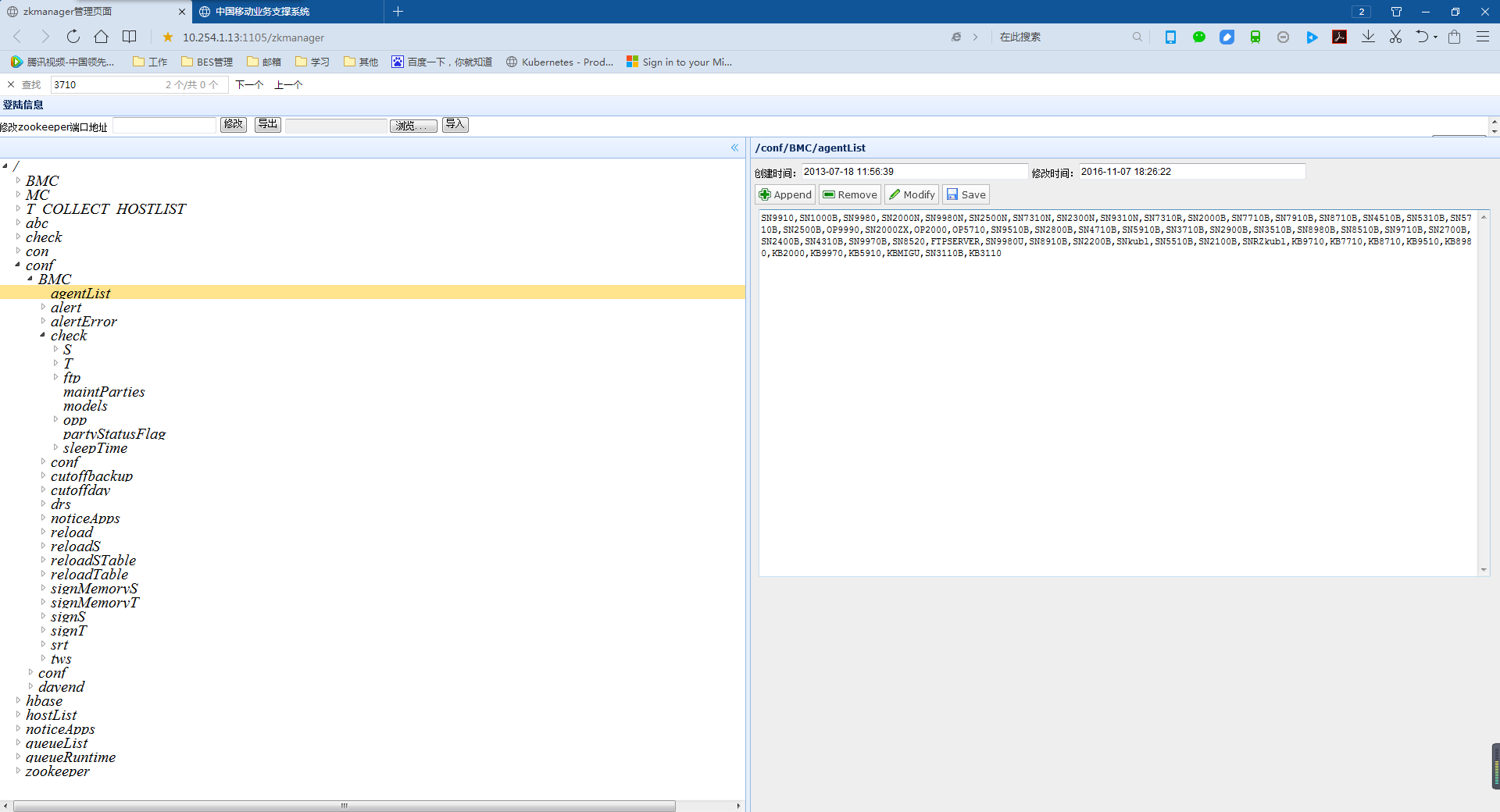


\*path内填写相应的node名字，value不要填写内容。

\*Save保存，cancle退出



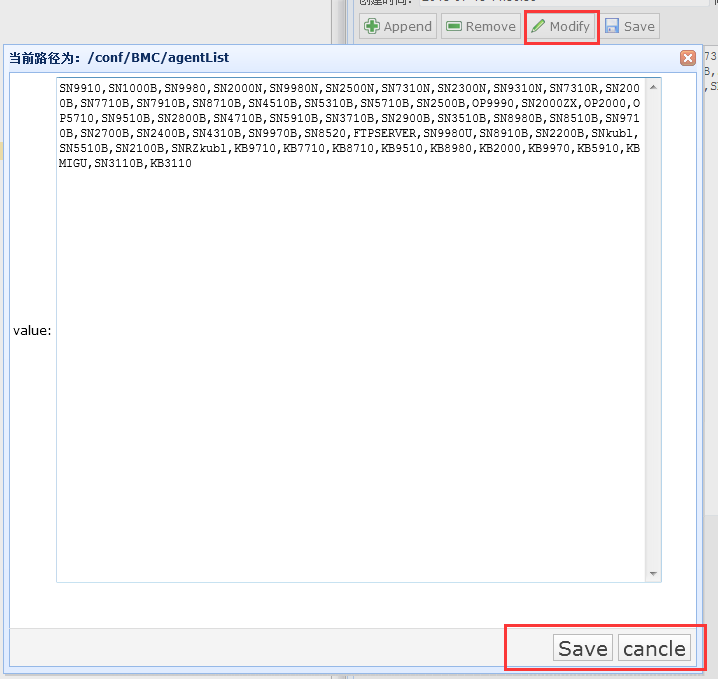
1. **/conf/BMC/agentlist增加相应的node名字**



\***/conf/BMC/agentlist增加node名字与/hostlist append 一定要相同**

\*右侧选择：Modify按钮 新增**node名字 格式 是 ,xxxxx**

\*Save保存，cancle退出

**3./conf/BMC/tws/exe\_para**

同3.2

### 3.4.2加密机后台配置

加密机security.conf文件下配置tws\_node.txt(如BMC中没有新增tws，此步骤不用操作)

1、登陆机器10.1.128.113 （BMCServer部署机器）

2、cd $ECIP\_HOME/BMC

ecip@cmcbnscs:/opt/cmcb/csc/ecip/BMC>ll

total 14064

drwxr-xr-x 2 ecip appadm 8192 Jul 10 21:45 bmcServerConf -----BMCServer配置文件目录

drwxr-xr-x 2 ecip appadm 16384 Aug 21 00:00 bmclogs ----BMCServer的日志目录

drwxr-xr-x 6 root sys 96 Sep 30 2014 c

drwxr-xr-x 5 ecip appadm 8192 Jul 25 2014 cutoff ----------日切程序

-rw-r--r-- 1 ecip appadm 7168000 Sep 26 2013 ftptest.tar

3、cd bmcServerConf

ecip@cmcbnscs:/opt/cmcb/csc/ecip/BMC/bmcServerConf>ll

total 336

-rw-r--r-- 1 ecip appadm 1809 Aug 11 20:25 config.conf ----BMC配置文件

-rw-r--r-- 1 ecip appadm 1543 May 12 2014 config.conf.20140512

-rw-r--r-- 1 ecip appadm 1472 Sep 26 2013 config.conf.bak

-rw-r--r-- 1 ecip appadm 1387 Sep 22 2013 config.conf.old

-rw-r----- 1 ecip appadm 53850 Sep 26 2013 ftptest.jar

-rw-rw-rw- 1 root sys 10764 Aug 13 12:18 tws\_node.xml --密钥申请，密钥配置文件下发配置文件

-rw-r--r-- 1 ecip appadm 9424 May 18 16:59 tws\_node.xml.20150518

-rw-r--r-- 1 ecip appadm 10505 Apr 28 16:00 tws\_node.xml.bak201504281600

-rw-r--r-- 1 ecip appadm 10203 May 21 12:06 tws\_node.xmlbak20150521

-rw-r--r-- 1 ecip appadm 10203 May 18 17:31 tws\_node20150518newmodified.xml

4、cat tws\_node.xml

...

<node>

<para name="host" value="SN5310B"/>

<para name="ip" value="10.254.66.17"/>

<para name="ftp\_type" value="SFTP"/>

<para name="port" value="22"/>

<para name="name" value="ecip"/>

<para name="pass" value="ecip"/>

<para name="file\_path" value="/hadoop/ecip/security/config/"/>

</node>

....

5、unix使用ftp，linux使用sftp 端口分别是21和22，fiel\_path和ip必须准确而且以/结尾。file\_path: env|grep TWSECU\_HOME

6、登陆http://10.1.128.113:1900/，admin/admin,重启节点aloneIns2（BMCServer）。

### 3.4.3重启agent节点

一般情况下，zkagent是部署在它所管辖的应用机器上的其中一台上，省内的通过查看加密机环境变量找到zkagent所在机器。（加密机和zkagent是必须待一起的，除非zkagent不需要参与tws功能）

1、登陆对应机器

2、cd $ECIP\_HOME

3、ps -exf|grep zk，查看是否是即所找的zkagent名字

4、cd zkAgent执行·Rstzk.persion 或者Rstzk.persion.linux。根据系统类型选择脚本执行。

5、等待

ecip@cmcscbks:/opt/cmcb/bks/ecip/zkAgent>Sending output to nohup.out

脚本运行完成

6、进入logs目录,tail -f zk\_agent.log,查看日志是否有报错。

7、同时ps -exf|grep zk查看zkagent是否启动成功。

# 测试小工具使用（庭鑫）

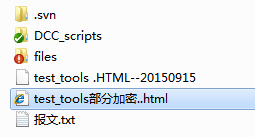
本地发送模拟报文，可以测试交易流程。通过应答报文查看具体问题。

测试工具：

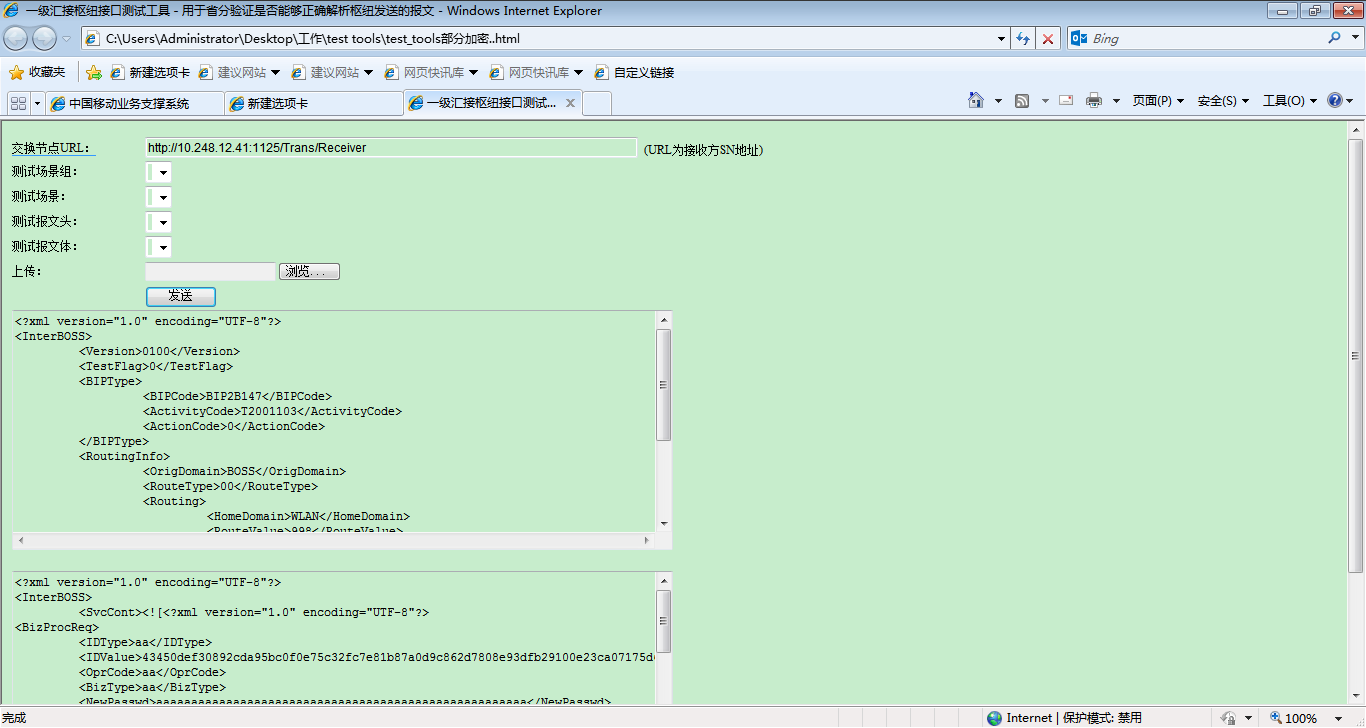
test\_tools部分加密.html

流程：

1. 使用IE浏览器打开测试工具test\_tools部分加密..html



2．打开后界面如下：



1. 需要填写：
2. 交换节点URL（上发网状网地址）

例如：http://10.255.242.71:1107/Trans/Receiver

1. 报文头

例如：

<?xml version="1.0" encoding="UTF-8"?>

<InterBOSS>

<Version>0100</Version>

<TestFlag>0</TestFlag>

<BIPType>

<BIPCode>BIP5A055</BIPCode>

<ActivityCode>T5000055</ActivityCode>

<ActionCode>0</ActionCode>

</BIPType>

<RoutingInfo>

<OrigDomain>UPMS</OrigDomain>

<RouteType>01</RouteType>

<Routing>

<HomeDomain>BOSS</HomeDomain>

<RouteValue>18202547473</RouteValue>

</Routing>

</RoutingInfo>

<TransInfo>

<SessionID>UPMSPID20170111020632659020</SessionID>

<TransIDO>TransIDO20170111020632659020</TransIDO>

<TransIDOTime>20170111020632</TransIDOTime>

</TransInfo>

</InterBOSS>

1. 报文体

例如：

<?xml version="1.0" encoding="UTF-8"?>

<InterBOSS>

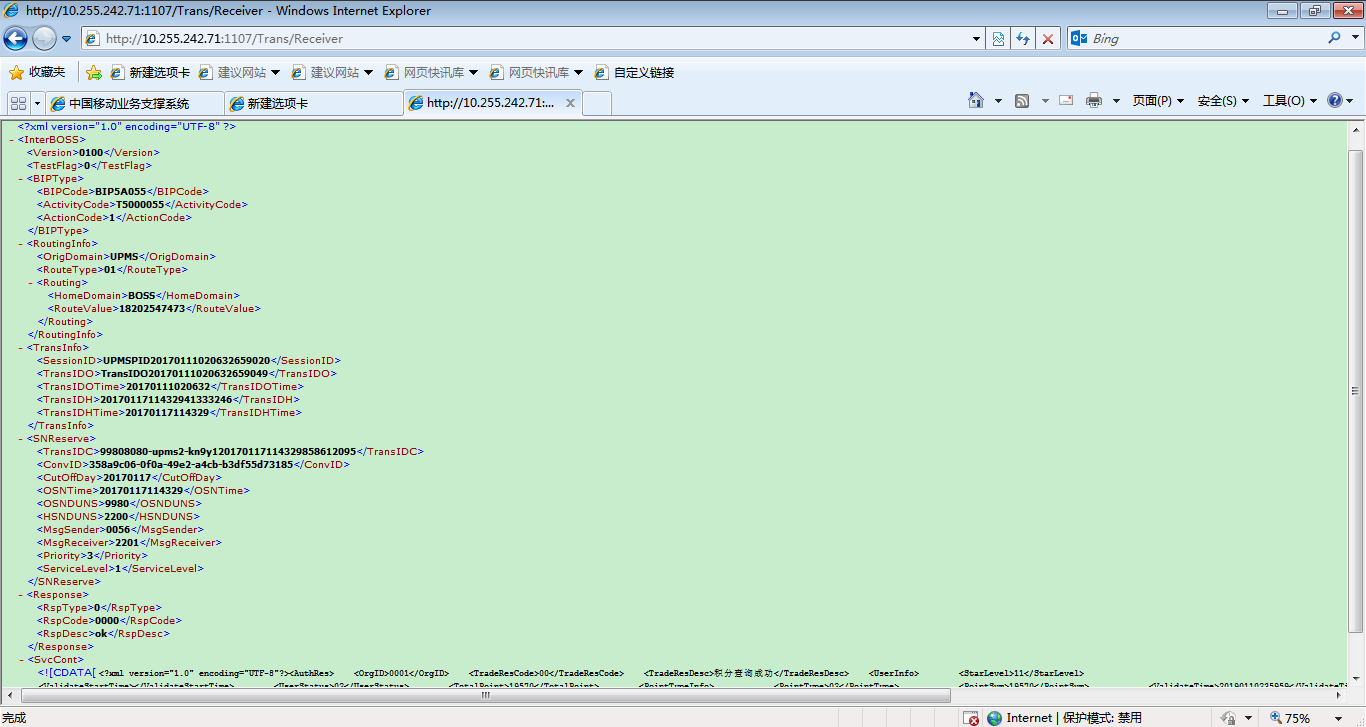
<SvcCont><![CDATA[<?xmlversion="1.0"encoding="UTF-8"?><AuthReq><OrgID>0001</OrgID><OrgName>0001</OrgName><MobileNum>18202547473</MobileNum><IdentCode>uad685a0953204820170111020614931</IdentCode></AuthReq>]]></SvcCont>

</InterBOSS>

测试系统发送报文，transido可以重复发送；生产系统发送报文transido每次发送需要修改，不可重复。

1. 填写完成后，点击“发送”按钮
2. 等待发送成功后，页面会返回应答报文，可以通过应答报文判断是否存在问题

应答界面如下：



# Osn与hsn的隔离

包括A10，F5操作

# 容灾