目录

[IGP采集常见问题解决方法汇总 2](#_Toc296673143)

[前言 2](#_Toc296673144)

[查找问题的主要思路 2](#_Toc296673145)

[需要了解的基本知识 5](#_Toc296673146)

[Java程序运行原理 5](#_Toc296673147)

[Java程序的内存和CPU占用 6](#_Toc296673148)

[连接数据库 6](#_Toc296673149)

[FTP原理 7](#_Toc296673150)

[sqlldr入库 11](#_Toc296673151)

[数据库方式采集出现的问题 11](#_Toc296673152)

[某个任务突然完全采不到数据了 11](#_Toc296673153)

[某个CLT表突然无数据或缺数据 12](#_Toc296673154)

[采集很慢 13](#_Toc296673155)

[采集到CLT表中的中文数据带有问号 13](#_Toc296673156)

[FTP方式采集出现的问题 13](#_Toc296673157)

[日志中显示FTP登录失败 13](#_Toc296673158)

[日志中显示listFiles失败 14](#_Toc296673159)

[日志显示MR文件下载成功，但文件却找不到 15](#_Toc296673160)

[采集入库后的数据和原始文件内容不同 15](#_Toc296673161)

[其它问题 16](#_Toc296673162)

[任务一直停留在一个时间点 16](#_Toc296673163)

[日志中始终显示select出来的任务数为0 16](#_Toc296673164)

[日志中始终显示有效任务数为0 16](#_Toc296673165)

[out.log、error.log被其它进程占用，导致IGP无法启动 16](#_Toc296673166)

[IGP在linux运行时，日志中显示”too many open files” 17](#_Toc296673167)

[LOG\_CLT\_INSERT表中的表名（CLT\_TBNAME列）为空 17](#_Toc296673168)

[原始数据生成延迟的问题 17](#_Toc296673169)

[注意事项 18](#_Toc296673170)

[config.xml配置项解释 18](#_Toc296673171)

[system节点 18](#_Toc296673172)

[db节点 20](#_Toc296673173)

[web节点 21](#_Toc296673174)

[alarm节点 23](#_Toc296673175)

[dataFileLifecycle节点 23](#_Toc296673176)

[datalog节点 24](#_Toc296673177)

[delayProbe节点 24](#_Toc296673178)

[traceFileter2Path节点 25](#_Toc296673179)

[sqlldr节点 26](#_Toc296673180)

[mail节点 26](#_Toc296673181)

[mr节点 26](#_Toc296673182)

[log4j节点 27](#_Toc296673183)

# IGP采集常见问题解决方法汇总

## 前言

此文档将现场使用IGP采集时，经常出现的问题总结出来，并给出解决问题的思路及方法。注意，这里说的只是与IGP采集有关的问题，即已经确定是CLT表里没数据、数据异常、缺数据、FTP文件下载不了等。如果是巡检发现的问题，请先将问题定位到CLT表部分，再参考此文档。

各位发现采集有问题时，请不要急于找研发，先借助这个文档，自己想办法排查问题，实现找不出问题了，再联系研发协助。

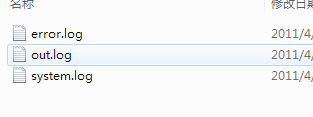
另外，请注意，排查问题，是技术工作，千万不要凭感觉来。比如，有些同事发现IGP下载不到某个原始文件了，我告诉他手工到FTP上看看有没有，然后他告诉我：“厂家FTP上有文件，很多，肯定有。”然而，我上去看的时候，发现FTP上文件是很多，但唯独没有我们需要的那一个。还有位同事，同样反映IGP下载不到某个文件了，手工到FTP上看，确实也有那个文件，但我连到现场看的时候，发现他手工连的和IGP任务表配的，不是同一个FTP地址。

我并不是说IGP程序没有问题，我想表达的是，排查问题一定要仔细一点，不要凭感觉走。比如查FTP上有没有文件，最基本的要求是：IP地址、端口号、用户名、密码，从任务表里查出来，不要说是哪个文档里有厂家接口信息，从文档里找的，或是说我的FlashFXP以前连过，有记录的；路径从IGP日志中打印的信息中来，手工连FTP的时候，在运行IGP的机器上连。

切记，一定要严谨。

## 查找问题的主要思路

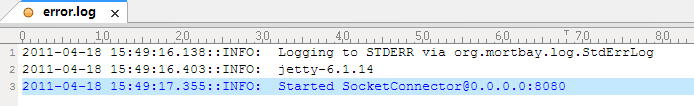
出现问题的主要思路，还是看IGP的日志，即IGP的log文件夹中的几个.log文件。日志有以下三种：



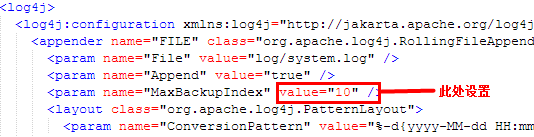
out.log和error.log，是对系统“标准输出”和“错误输出”的重定向，说白了，就是我们在windows的cmd窗口或unix的终端上看到的打出的信息，在IGP运行时，为了避免占用操作系统的内存及CPU（实际情况发现，在控制台打印大量信息时，CPU和内存的占用会非常大），将其重定向到了文件中。

out.log不用去管他，因为system.log中全部都有了，而out.log中的内容是不全的，只有INFO级别以上的日志记录在了out.log里（避免out.log太大，所以设置了，至于日志的INFO、DEBUG、ERROR等级别的意思，请GOOGLE或百度一下）。

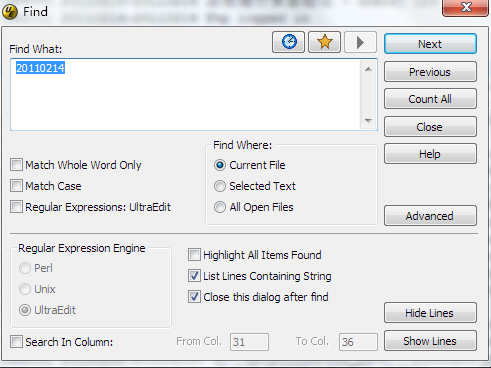
error.log中，如果有且只有下面这三行内容，也不用管它，它是一个第三方组件打印出的内容，只是刚好打印在了系统错误输出端。如果出现了其它内容，通常就是较严重的问题了，例如内存溢出。



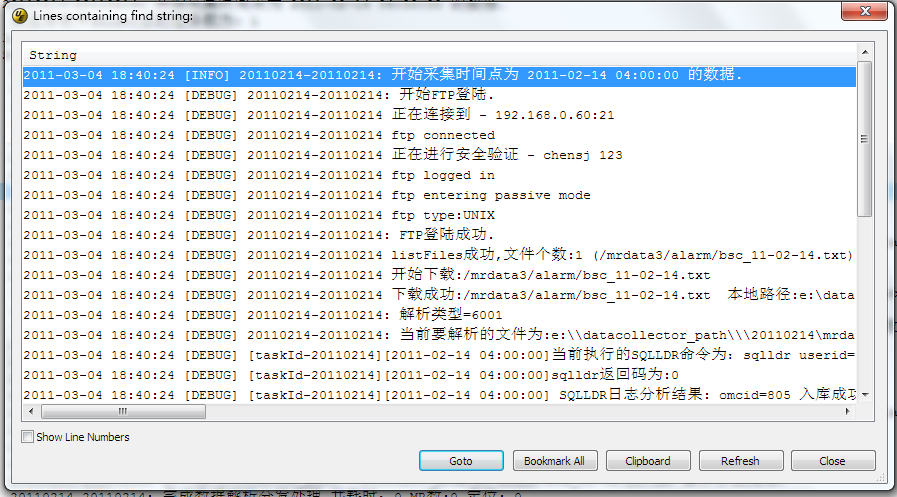
system.log是10MB一个，一共十个，滚动生成的，system.log是最新的，即当前在打印的日志。system.log.1、system.log.2等，则是滚动备份的日志，后面的数字越大，就表示是越老的日志。日志保存的个数，可以在config.xml中设置。

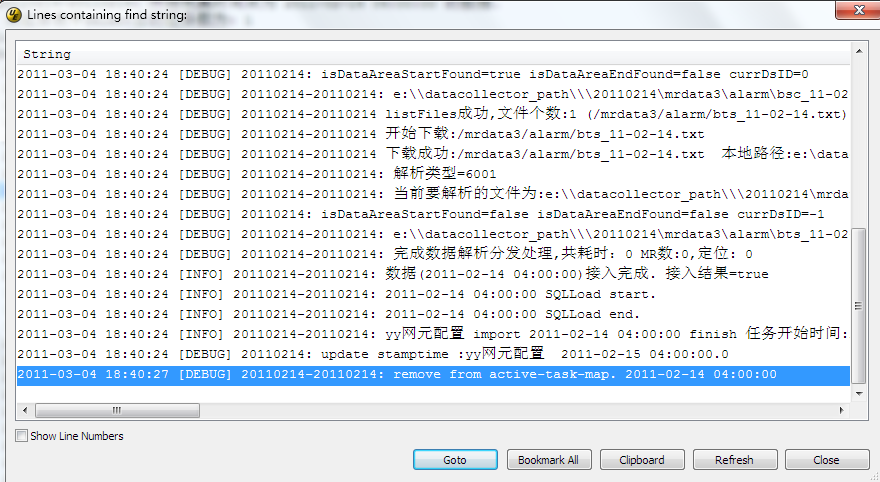


某个采集出现问题时，首先要知道有问题的采集，在igp\_conf\_task表中的任务号（task\_id）是多少。拿到任务号，在system.log中搜索。通常是用UltraEdit打开日志，然后ctrl+f打开搜索对话框，输入任务号，选择”List lines containing string”，然后点击”Next”开始搜索（具体操作，不同的UltraEdit版本，可能有些不同，请GOOGLE或百度一下）。



搜索完成之后，将显示出指定任务所打印出的所有信息。





也许您并不知道IGP的原理，但从日志可以看出，一个任务在一个时间点的流程，是以“开始采集时间点为XXX的数据”开头，以“remove from active-task-map”结束的。那么在这两段文字之间的内容，便是一个任务在一个时间点的所有信息了。如果在system.log搜索出来的不全，可能是因为日志跨文件记录了，您可以将system.log.1、system.log.2等日志文件中的内容，按顺序合并到一起，然后再搜索。

日志中记录的任务号，有些是“20110214-20110214”这样的格式，如果“-”两边的数字相同，都是igp\_conf\_task表中的任务号，那么，表示这是一个正常任务；如果“-”左边的是任务号，右边却是个不同的数字，这表示它是一个补采任务（即igp\_conf\_rtask表中的任务）。

## 需要了解的基本知识

### Java程序运行原理

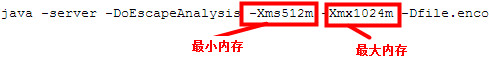
Java程序是运行在JVM（JAVA虚拟机）上的，所以Java程序可以跨平台。一个Java程序运行时，我们在任务管理器中看到的是一个”java.exe”进程，这个”java.exe”就可以理解为是JVM.

举个例子，.doc文件是word文档，而word是打开.doc文件的工具，如果这个word有多个操作系统的版本，那么，我在多个操作系统上，都可以打开.doc文件。

相对来说，Java程序是个.class文件，而JVM就是打开这个.class文件的工具，然而，JVM有各种各样的版本，WINDOWS、LINUX、AIX、64位、32位等等都有，所以，无论在哪个平台，都可以使用。

### Java程序的内存和CPU占用

先说内存，一个java程序用多少内存，是在启动时通过参数指定的。在igp.bat或igp.sh中可以看到以下内容。



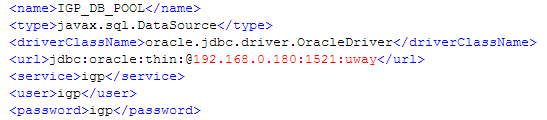
比如最大内存指定的是1024M，那么程序一启动，不管有没有干活，带起这个程序的”java.exe”进程马上就会占用1024M的内存。而当程序真正占用的内存超过了1024M，因为内存超过限制而无法干活的线程，就会报错，这时在error.log中可看到报错信息。如果是主线程报错，程序就退出了；如果是某个任务线程报错，那么，这个任务就中断了。

要注意的是，在WINDOWS，即使是64位的，最大内存一般也只能设到2048M，再设大了，程序就启动不了了，在error.log中可以看到报错信息。而在基于UNIX的操作系统上，只要机器内存够大，”-Xmx”设到多大都可以。

关于CPU的占用。由于Java程序运行在JVM上，JVM已经将各种与操作系统的交互做到很好了，所以一般不会占用过高的CPU，除非是Java程序写得太不合理。

### 连接数据库

Java程序连接数据库，有以下五个参数要设置，这里就对应config.xml中连接我们自己的Oracle数据库来讲。



1. *type：这是建立连接池的数据源接口，这个定死了，就是”javax.sql.DataSource”，不用管它；*
2. *driverClassName：这是连接数据库的驱动，各个数据库厂家都不同，对于Oracle数据库，定死了，就是” oracle.jdbc.driver.OracleDriver”，不用管它；*
3. *url：连接数据库的描述符，以thin方式连接oracle数据库，就是” jdbc:oracle:thin:@数据库IP:数据库端口:数据库服务名”这样的格式，看起来有点复杂，不过就是这样规定的，不用问我为什么。但是，通过这一串东西，我们还是能看出数据库的IP及端口信息。注意，最后那个是“数据库服务名”（通过” select instance\_name from v$instance”查出来的），而不是Oracle客户端配的tns名。*
4. *user：Oracle用户名；*
5. *password：Oracle密码。*

以上就是Java连接数据库的五个参数，上图中的name，只是个连接池的名字，写成什么都无所谓。service是Oracle客户端中设置的tns名，作用下面再讲。

Java程序连接数据库有几种方式，这里只讲IGP用过的两种，thin方式和oci方式。但无论哪种方式，都是需要上面说的五个参数，只是格式和内容有所不同。

1. *oci方式：需要机器上装了Oracle客户端才行，它是通过本地dll/so驱动的。*
2. *thin方式：机器上不需要安装Oracle客户端，纯Java实现，上图中的设置，就是thin方式的。*

理论上讲oci方式要比thin方式效率高一些，但IGP连接我们的Oracle数据库，只是查一下任务表、采集模板表等，看不出效率上的区别。而且，如果用oci方式，配置较为复杂。所以，推荐使用thin方式连接我们的Oracle数据库。

各位会发现，虽然使用的thin方式连接，但部署时还是要求安装Oracle客户端，这是因为数据入库时，是使用的sqlldr，而sqlldr是安装了Oracle客户端才有的工具。执行sqlldr，需要知道tns名，所以，上图中有个”service”来配置tns名。

### FTP原理

#### 概述

FTP方式采集，现场提出的问题特别的多，所以，这里简单讲一下FTP的工作原理。

有一点要说明一下，平时我们用windows自带的FTP命令行工具来连接某个FTP，会使用一些命令，比如dir、ls、pwd、cd等，这些命令，并不是FTP协议所规定命令，而是“windows的FTP工具的命令”。Windows的FTP命令行工具，只是像FlashFXP这样的工具，只是它没有窗口界面，只能靠命令来驱动。但它和FlashFXP这样的工具一样，帮我们执行了FTP协议规定的一些比较复杂的过程，而我们只需要看结果。

与FTP服务器交互，有两种模式，主动模式（PORT）和被动模式（PASV）。主动和被动，是以FTP服务器为对象来讲的，主动模式是FTP服务器主动发送数据到客户端的某个端口，被动模式是FTP服务器被客户端从某个端口取数据。

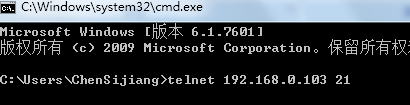
注意，这里说的“数据”，是除了FTP命令之外的所有东西，包括下载/上传的文件，及LIST出来的文件列表等。FTP命令，走是的平常所说的FTP的21端口。

下面讲一下两种模式的交互过程。前面说了WINDOWS的FTP工具只是个工具，并不能体现FTP协议所指定的命令。下面的例子，用telnet方式连接FTP服务器，并执行一个LIST命令（即要求FTP服务器给出当前目录的文件列表，也就是WINDOWS的FTP工具的dir命令完成的工作），可以大致看出FTP的工作方式。

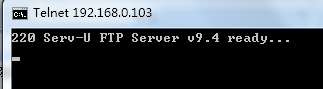
*注意：下面例子中，FTP服务器是本机建的，即客户端和FTP服务器的IP是一样的（192.168.0.103）。*

#### 主动模式

**第一步，** telnet到FTP服务器的端口；



**第二步，**按下回车，可以看到如下内容（FTP服务器的每个响应信息，最前面都会有一个三位的数字，1和3开头的，表示中间应答，也就是说FTP服务器在等待客户端的下一个动作；4和5开头的，表示失败，或者拒绝；2开头的表示成功完成。三位数字后面的内容，每个FTP服务器可能不同。具体的内容，请GOOGLE或百度一下。）；



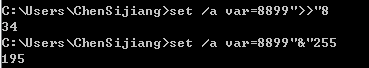
**第三步，**执行user和pass命令，进行登录；



**第四步，**执行port命令，告诉FTP服务器，往我们的8899端口上送数据；



*port后面的”192,168,0,103,34,195”，表示我们的IP和端口信息，是以逗号分隔的。IP部位我们可以看出来，是”192.168.0.103”。后面的”34,195”，是表示8899端口。计算方法是“端口号>>8,端口号&255”，自己计算比较麻烦，可以通过windows的cmd位运算，来计算，如下图：*

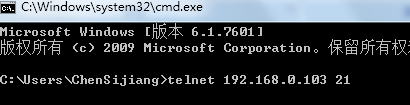
**

**第五步，**执行LIST命令，获取当前目录的文件列表。可以看到，FTP服务器最后返回了一个”425 Cannot open data connection.”的错误，原因是我们刚才指定了接收数据端口是8899，而我们并没有程序开启8899端口来接收FTP服务器发送的数据。

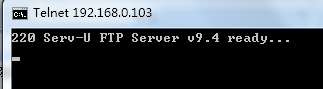


#### 被动模式

**第一步，** telnet到FTP服务器的端口；



**第二步，**按下回车，可以看到如下内容（FTP服务器的每个响应信息，最前面都会有一个三位的数字，1和3开头的，表示中间应答，也就是说FTP服务器在等待客户端的下一个动作；4和5开头的，表示失败，或者拒绝；2开头的表示成功完成。三位数字后面的内容，每个FTP服务器可能不同。具体的内容，请GOOGLE或百度一下。）；



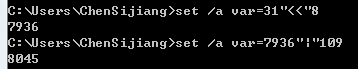
**第三步，**执行user和pass命令，进程登录；



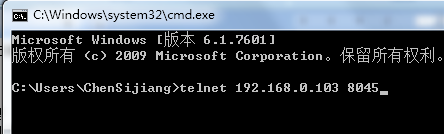
**第四步，**执行pasv命令，让FTP服务器告诉我们，要从它的哪个端口取数据。之后再执行LIST命令，FTP服务器返回信息，表示已经打开刚才显示的端口，供客户端连接上来取数据；



*可以看到FTP服务器返回的信息中有”192,168,0,103,31,109”这么一串数字，可以看出IP是192.168.0.103；端口是8045，是从”31,109”算出来的，算法为“(第一个数字<<8)|第二个数字”，自己计算比较麻烦，可以通过windows的cmd位运算，来计算，如下图：*

**

**第五步，** 我们新开一个CMD窗口，通过telnet命令行工具试试192.168.0.103的8045端口是不是通的。通过telnet来试，如果没有报“无法连接XX端口”之类的错误，则表示，FTP服务器给的8045端口是通的。如果连不上8045端口，则说明FTP服务所在的机器，有防火墙限制了端口连接，也有可能是我们自己的机器限制了连接。当然，即使连上，看到的也是一片空白，并没有实际内容，这是正常的，因为FTP服务器发送数据内容时，并没有进行telnet选项协商，所以windows的telnet工具看不到内容，这是正常的。我们只要确定端口是通的，没有报错，就可以了。



### sqlldr入库

system.log中打印了sqlldr入库时的返回码，请在日志中搜索“sqlldr返回码为”或“ret=”。这里所说的“返回码”，严格的说，是sqlldr进程的退出码。退出码的含义如下：

*0：入库成功；*

*1：未能连接到ORACLE数据库；*

*2：部分入库成功（部分数据因违反约束，类型不匹配等未能入库）；*

*3：表不存在、列不存在等完全无法入库的错误。*

在日志中搜索“sqlldr日志=”，可以找到sqlldr日志文件的位置，我们可以打开sqlldr日志，分析出现的问题。

## 数据库方式采集出现的问题

### 某个任务突然完全采不到数据了

首先，要确定厂家的数据库是否能正常访问。我们通过以下数语句，查出厂家数据库连接方式。

select t.dburl as "数据库连接信息",

d.host\_user as "数据库用户名",

d.host\_pwd as "数据库密码"

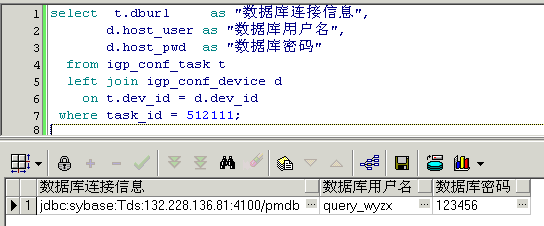
from igp\_conf\_task t

left join igp\_conf\_device d

on t.dev\_id = d.dev\_id

where task\_id = 任务号;

例如下图，我们查出了一个厂家的数据连接信息。



于是，我们可以知道，厂家是sybase数据库，IP是132.228.136.81，端口是4100，数据库名是pmdb，用户名是query\_wyzx，密码是123456.然后，我们用”Aqua data studio”之类的数据库工具，连接上去。

*注意，不要用数据库的官方工具来连，因为通常官方工具通常可配置项比较少，比如sqlserver的”Management Studio”工具，默认是连接1433端口，没法改变，而很多厂家并没有用默认端口。即使可以改变端口，也不要使用它，因为用第三方工具，比较能够反映出客观情况。*

现在看一下，通过我们查出的信息，能否正常连接上去，并打开指定数据库，进行数据查询。

### 某个CLT表突然无数据或缺数据

首先，通过[上面的方法](#_某个任务突然完全采不到数据了)，连接到厂家库。

然后，需要知道有问题的CLT表对应厂家库的哪张表。 执行以下语句：

select t1.tempfilename as "模板文件名"

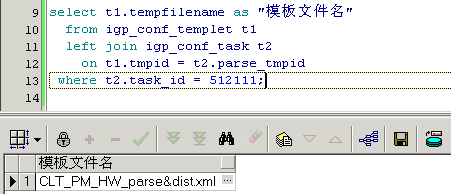
from igp\_conf\_templet t1

left join igp\_conf\_task t2

on t1.tmpid = t2.parse\_tmpid

where t2.task\_id = 任务号;

如下图，我们查出了这个采集任务的解析模板名：



然后，在对应的采集机上，找到IGP程序所在目录，找到conf目录下的config.xml文件，打开它，找到” <templetFilePath>”节点，这个节点里填写的是一个目录。接着，进入到这个目录，打开文件名和刚才查到的解析模板名一样的文件，搜索一下CLT表名，就能在同一行发现对应的厂家表名了。

现在只是知道厂家表名，还不知道IGP采集时，是执行的怎样的语句。要想知道，也可以在模板中找出。还有一个更简单的方法，在system.log日志中搜索厂家表名，就可以发现，程序打印出了在厂家库执行的select语句。请将select语句拿出来，在厂家库上执行，对比CLT表中入库的数据，如果不一样，请在system.log中搜索clt表名，您可以看见带有“sqlldr日志分析结果”字样的行，这行最后，有个“sqlldr日志=”，后面是sqlldr入库日志文件的路径，请依此打开sqlldr日志，分析入库失败原因。

### 采集很慢

原因有多种，解析过程慢，入库过程慢等。

1. sqlldr入库无报错，但很慢，这需要手工测试；
2. sqlldr入库时，量数据违反唯一约束，这会导致入库非常慢，请分析数据，看是否索引建得不对，或者规则要改了；
3. 在厂家库上查询数据很慢，这需要手工连到厂家库上实验。解决方法，一是将采集任务拆分成多条，二是想办法改善在厂家库上查询数据的速度。

### 采集到CLT表中的中文数据带有问号

通常是因为厂家库的字符集和IGP程序使用的字符集不同，将打开igp.bat/igp.sh启动脚本，找到下图中标出的内容。将”gb2312”改成”gbk”或”utf-8”，进行重启采集进行尝试。



## FTP方式采集出现的问题

### 日志中显示FTP登录失败

首先请通过以下语句，查询出厂家FTP连接信息。

select d.host\_ip as "FTP地址",

t.dev\_port as "FTP端口",

d.host\_user as "FTP用户名",

d.host\_pwd as "FTP密码"

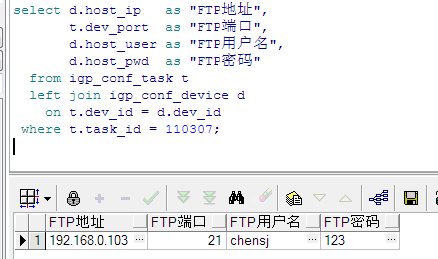
from igp\_conf\_task t

left join igp\_conf\_device d

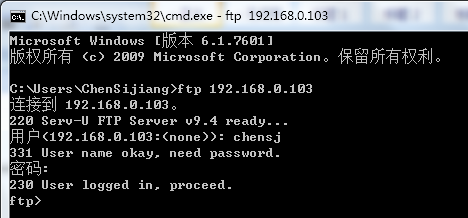
on t.dev\_id = d.dev\_id

where t.task\_id = 任务号;

执行完毕，结果如下：



首先确定这些列中的数据，没有多余的回车换行符。然后，用查询出的连接信息，在采集机上，通过系统自带的FTP命令行工具连接到厂家FTP，看能否正常登录上去。连接成功，并且在输入用户名和密码后，FTP返回”2XX”开头的信息，则说明登录没有问题，否则，FTP连接信息不正确。

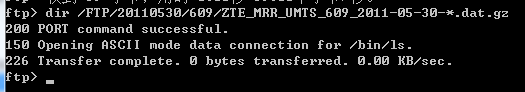


### 日志中显示listFiles失败

首先，用[上面的方法](#_日志中显示FTP登录失败)，登录到厂家FTP.

然后，在日志中找出一个listFiles失败的路径的例子，如下图，” /FTP/20110530/609/ZTE\_MRR\_UMTS\_609\_2011-05-30-\*.dat.gz”便是listFiles失败的路径。

在FTP上执行“dir 路径”的命令，看返回的结果。如果是象下图这样，说明文件不存在。



如果象下图这样，有文件列表出现，则表示文件存在。



*注意，不同的FTP服务器，返回的信息，不一定相同，还需要自己观察FTP返回的信息内容。*

导致listFiles失败的原因有很多种，具体可以[按照前面介绍过的方法](#_FTP原理)，更深入的测试一下。比如有可能是FTP服务器限制了端口（IGP使用被动模式进行FTP交互），或者不支持带有通配符的路径等。

### 日志显示MR文件下载成功，但文件却找不到

IGP的config.xml配置文件中的” <currentPath>”指定的目录，即FTP文件下载目录基准目录。然后，下载下来的文件，会按FTP上的目录结构存放。具体一个文件的位置，请在日志中搜索“下载成功”，搜索出来的行，有“本地路径:”这么一段，这里便有下载下来的文件在本地磁盘的绝对路径。



### 采集入库后的数据和原始文件内容不同

通常是因为原始文件在IGP采集之后，又被修改了。原始文件在什么时候被采集的，在IGP的system.log日志中搜索文件名即可分析出来，至于FTP上的原始文件什么时候被更新过，请参考[原始数据生成延迟的问题](#_原始数据生成延迟的问题)。

## 其它问题

### 任务一直停留在一个时间点

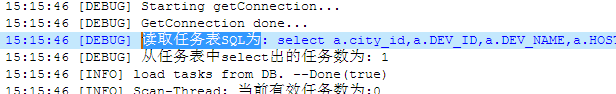
最常见的原因，igp\_conf\_task表被锁。现场人员修改任务表时，修改之后，却忘了解锁或提交，导致IGP无法将任务的时间点修改到下一时间。请在pl/developer的sessions中查看锁表情况。

### 日志中始终显示select出来的任务数为0

首先确定config.xml配置的数据库连接信息是否正确。然后，确定以下几点：

1. 任务的collector\_name是否设置正确；
2. 任务的isused列是否是1；
3. igp\_conf\_task、igp\_conf\_templet、igp\_conf\_device表是否正确关联。

比较简单的方法，在日志中搜索“读取任务表SQL为”，即可得到IGP查询任务时所用的SQL语句，将它拿出来，在pl/developer中执行，可以推断出原因。



### 日志中始终显示有效任务数为0

首先确定config.xml配置的数据库连接信息是否正确。然后，确定以下几点：

1. 采集机的系统时间是否正常；
2. 检查igp\_conf\_task表的suc\_data\_time、end\_data\_time、collect\_timepos列。

### out.log、error.log被其它进程占用，导致IGP无法启动

通常是因为上次退出IGP时，sqlldr占用了系统输出端未释放。可以通过结束所有”sqlldr.exe”进程来解决，如果无法解决，又找不到占用out.log、error.log的进程，请下载一个名为”unlocker”的小工具（有32位和62位版本），可以查看文件/目录的占用进程，并解除占用。

### IGP在linux运行时，日志中显示”too many open files”

基于unix的操作系统，限制了一个用户的最大进程数，每个进程的最大打开文件数（unix下的文件并不仅仅表示磁盘上的文件，所有资源都称作“文件”，例如网络连接）等，请用”ulimit -a”命令查看。

解决方法如下：

1. 用root用户登录，打开”/etc/sysctl.conf”文件（请注意备份），在最下面加上以下附件中的内容（共一行内容）并保存；



1. 打开”/etc/security/limits.conf”（请注意备份），在最下面加上以下附件中的内容（共四行内容）并保存；



1. 重启采集机，重启后，用运行IGP的用户登录，执行"ulimit -a"命令，会看到一个列表，显示了各种参数。请观察以”open files”开头的这一行的最后，是不是”65536”，如果是，说明修改成功，否则说明有问题，请立即反馈。

注意：请用vi命令以进行以上操作，不要用FTP下载下来改，并千万要小心别在文件中加入多余的字符（包括空格、换行符等）。

### LOG\_CLT\_INSERT表中的表名（CLT\_TBNAME列）为空

通常是因为SQLLDR日志解析模板和采集机上安装的ORACLE客户端版本不匹配。SQLLDR日志解析模板即IGP的CONF目录中的” SqlLdrLogAnalyseTemplet.xml”文件，它是用来告诉IGP如何解析SQLLDR生成的日志。LOG\_CLT\_INSERT表中的入库成功条数和表名，是从SQLLDR日志中解析出来的，而SQLLDR生成的日志，根据ORACLE客户端的版本不同，内容也有不一样。但大致分为中文版和英文版，在控制台执行”sqlldr”命令，如果显示出来的帮助信息是中文的，就是中文版。

以下是中文版和英文版的SQLLDR日志分析模板内容，请根据实际情况替换，替换后保存，重启程序生效。

### 原始数据生成延迟的问题

无论是数据库方式还是FTP方式采集，都有这个问题。FTP方式采集，登录到FTP，并观察文件的最后修改时间即可。数据库方式通常要靠平时的观察来得出结论，但华为M2000数据库的性能表中，通常有INSERTTIME这么一列，它是记录数据入库时间的，作用类似于FTP上的文件最后修改时间。

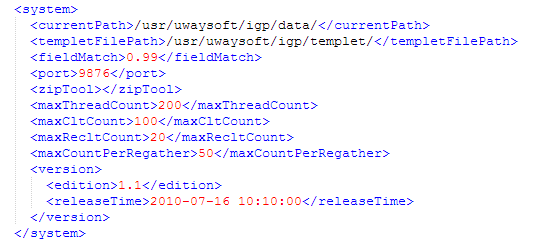
## 注意事项

1. 如果要修改任务时间，请在IGP关闭后再修改；
2. 修改任务表后，一定要记得解锁；
3. 同一个IGP，不要开两个，如果一台机器上要运行两个IGP，请单独拷贝一份出来，并进行相应的配置。

## config.xml配置项解释

### system节点

IGP常规配置。



#### currentPath

IGP临时文件存放目录。厂家原始文件 、SQLLDR临时文件等，均被放在此目录中。

#### templetFilePath

采集模板的存放位置。IGP从此处读取采集模板。

#### fieldMatch

未使用。

#### port

采集端口。telnet方式管理IGP程序时，IGP所使用的侦听端口。

#### zipTool

IGP运行在Windows平台时，Winrar解压工具的安装路径。在非Windows平台使用IGP时，此配置项不使用。

#### maxThreadCount

未使用。

#### maxCltCount

最大同时执行的采集任务数量（IGP\_CONF\_TASK表中配置的任务）。

#### maxRecltCount

最大同时执行补采任务数量（IGP\_CONF\_RTASK表中的任务）。

#### maxCountPerRegather

从IGP\_CONF\_RTASK表中查询补采任务时的最大select行数。

#### edition

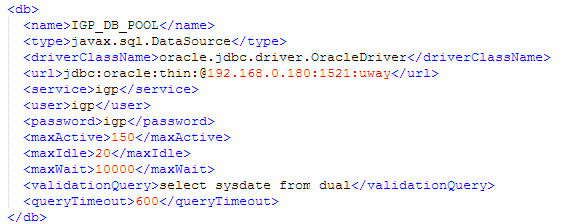
固定值，请勿更改。

#### releaseTime

固定值，请勿更改。

### db节点

数据库操作相关配置。



#### name

固定值，请勿更改。

#### type

固定值，请勿更改。

#### driverClassName

固定值，请勿更改。

#### url

连接我方Oracle数据库的数据库连接字符串。格式为“jdbc:oracle:thin:@**数据库IP**:**数据库服务端口**:**数据库服务实例名**”（加粗红色字体部分是需要修改的，其余固定不变）。

#### service

采集服务器安装的Oracle客户端所配置的我方数据库别名。

#### user

我方数据库用户名。

#### password

我方数据库密码。

#### maxActive

我方数据库连接池配置，最大活动连接数。

#### maxIdle

我方数据库连接池配置，最大空闲连接数。

#### maxWait

我方数据库连接池配置，当数据库连接池的活动连接数已达到最大时的最大等待连接时长（毫秒）。如果超过此时间，操作线程将会产生异常。

#### validationQuery

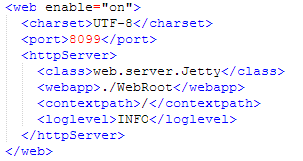
固定值，请勿更改。

#### queryTimeout

在厂家库上执行select语句的超时时间（秒）。如果超过此时间，操作线程将产生异常。

### web节点

IGP的web界面相关配置。



#### charset

在进行http请求与响应处理时的字符集。

#### port

web服务的侦听端口。

#### class

固定值，请勿更改。

#### webapp

固定值，请勿更改。

#### contextpath

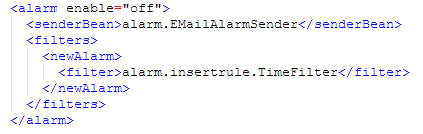
固定值，请勿更改。

#### loglevel

固定值，请勿更改。

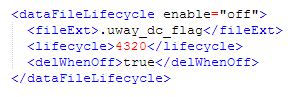
### alarm节点

暂不使用，请勿更改其中内容。



### dataFileLifecycle节点

原始文件生命周期管理相关配置。



#### enable

是否启用原始文件生命周期管理。“off”为不启用，“on”为启用。

#### fileExt

固定值，请勿更改。

#### lifecycle

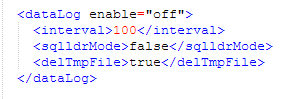
原始文件在本地保存的时间（分钟）。

#### delWhenOff

在未启动原始文件生命周期管理时，是否在采集完成之后删除原始文件。

### datalog节点

数据库方式的采集日志配置（采集日志记录在IGP\_DATA\_LOG表）。



#### enable

是否启用数据库方式的采集日志。“off”为不启用，“on”为启用。

#### inerval

日志提交间隔条数。即在内存中累积多少条日志后，向数据库提交。

#### sqlldrMode

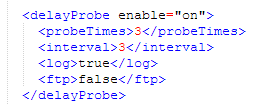
是否以sqlldr方式将日志记录到数据库。“true”为使用sqlldr方式，“false”为使用insert方式。

#### delTmpFile

使用sqlldr方式记录数据库日志时，是否删除sqlldr临时文件。“true”为删除，“false”为不删除。

### delayProbe节点

探针功能相关配置。



#### enable

是否启动探针功能。“on”为启用，“off”为不启用。

#### probeTimes

确定数据平稳的探测次数阀值。即连续探测到多少次数据不变时，确定数据已经平稳了。

#### interval

探测间隔时长（分钟）。

#### log

是否记录探测日志。

#### ftp

是否对FTP方式的采集任务使用探针功能。

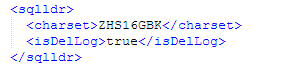
### traceFileter2Path节点

C网GPS定位日志转换工具的绝对路径。



### sqlldr节点

采集任务执行sqlldr入库时的相关配置。



#### charset

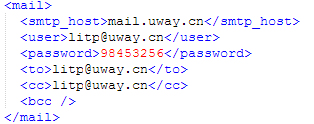
执行sqlldr时所使用的字符集。

#### isDelLog

是否在入库完成后删除sqlldr临时文件。“true”为删除，“false”为不删除。

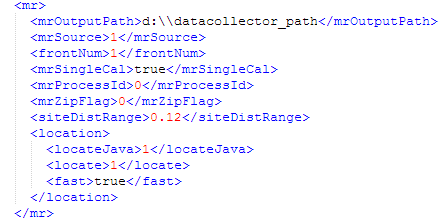
### mail节点

暂不使用，请勿更改其中内容。



### mr节点

mr处理相关配置。除mrProcessId外，其余配置均暂不使用，请勿更改。



#### mrProcessId

IGP程序多开标识。设置为“0”时，表示不使用。配置为非“0”值时，IGP认为采集服务器的机器名为“采集服务器机器名@mrProcessId”，以便在同一采集服务器上运行多个IGP程序，使各IGP程序的任务扫描不冲突。

### log4j节点

IGP系统日志相关配置。

