HiBI数据推送工具部署指南

(仅供内部使用）

For internal use only

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 拟制:  Prepared by | 汤清正 |  | 日期：  Date | 2012-11-11 |
| 审核:  Reviewed by |  |  | 日期：  Date | yyyy-mm-dd |
| 批准:  Granted by |  |  | 日期：  Date | yyyy-mm-dd |



华为技术有限公司

Huawei Technologies Co., Ltd.

版权所有 侵权必究

All rights reserved

修订记录Revision record·

|  |  |  |  |
| --- | --- | --- | --- |
| 日期  Date | 修订版本Revision version | 修改描述  change Description | 作者  Author |
| 2012-11-11 | V0.9 | 初稿 | 汤清正 |
| 2012-11-27 | V1.0 | 增加对SqlSever数据源的支持 | 李学智/00194471 |
|  |  |  |  |

[1 概述 4](#_Toc340741479)

[2 部署 4](#_Toc340741480)

[2.1 前置条件 4](#_Toc340741481)

[2.2 部署步骤 4](#_Toc340741482)

[3 配置 4](#_Toc340741483)

[3.1 环境配置 4](#_Toc340741484)

[3.2 业务配置 7](#_Toc340741485)

[3.2.1 数据库类推送任务 7](#_Toc340741486)

[3.2.2 文件类推送任务 8](#_Toc340741487)

[3.2.3 目标文件格式配置 14](#_Toc340741488)

[3.2.4 传输选项配置 14](#_Toc340741489)

[3.3 日志配置 15](#_Toc340741490)

[4 运行 15](#_Toc340741491)

[4.1 自动调度 15](#_Toc340741492)

[4.2 手动执行 16](#_Toc340741493)

[4.3 日常维护 17](#_Toc340741494)

[5 附录 17](#_Toc340741495)

[5.1 SSH密钥认证 17](#_Toc340741496)

[5.2 FAQ 18](#_Toc340741497)

# 概述

数据推送工具用于采集业务数据，并将其推送到BI环境。主要特点：

* 支持多种数据源，包括：Mysql、Oralce、日志文件、HIVE（暂未测试）
* 丰富的配置项，包括数据源、提取规则、目标文件、传输等的配置
* 支持配置多个采集任务，并发运行
* 支持网络传输失败多次重试

# 部署

## 前置条件

1. Java

推送工具采用java语言写就，故要确保java运行环境正常。安装合适的JDK，版本要求在1.6及以上。

1. SSH

推送工具采用SCP 、SSH远程传输文件、远程执行脚本，故要确保SSH服务开启并能保持建立长连接。

SSH长连接配置步骤：

修改/etc/ssh/sshd\_config文件，将ClientAliveInterval设置为60，ClientAliveCountMax设置为3.

重启sshd服务

service sshd restart

## 部署步骤

1. 创建用户

可以为推送工具创建1个单独的操作系统用户，比如hibi；也可以与其他业务合用1个用户。

1. 软件包上传

将软件包DataPush.zip上传至*$HOME/hib*i目录下，并解压。

# 配置

## 环境配置

环境配置文件EnvConf.xml用于配置数据源、传输服务等运行环境信息。包括：

* 部署主机节点

通常配置为IP，作为提取到的目标数据文件名的前缀

* TCC监听地址

上报告警事件用，预留，功能未实现

* 各种数据源

支持Mysql、Oracle、文本文件（如日志文件）的数据源配置，支持配置多个数据源

* 远程传输服务

支持SSH密码、密钥两种登录认证方式，支持配置多个传输服务。

推荐采用密钥认证方式。公司安全基线规定，密码必须两个月更改一次，采用密码方式传送数据时，配置文件需要定期更新，增加了额外维护工作。关于密钥认证方式可咨询各自的主机管理员或网络管理员，或参考附录《SSH密钥认证》。

EnvConf.xml配置项：

|  |  |  |
| --- | --- | --- |
| 配置项名称 | 含义 | 可能取值 |
| Node | 部署主机节点 | 本地IP |
| TCC | TCC监听 | IP + 端口 |
| DBSource | 数据库数据源 | 数据库类型、IP、端口、数据库名称、用户名、密码、追加参数 |
| FileSource | 文件源 | 文件源路径 |
| TransportService | 传输服务 | IP、端口、用户名、登陆方式、目标路径等 |

典型配置示例如下（红色字体部分通常是需要配置的）：

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

<Conf>

<!-- 部署主机节点 -->

<Node>10.120.5.47</Node>

<!--

TCC监听

上报告警事件用，预留，功能未实现

-->

<TCC>

<Host>10.120.5.45</Host>

<Port>8888</Port>

</TCC>

<!--

数据库数据源，支持myql/oracle

可多个，以id唯一标示，用户根据需要自行配置

-->

<!-- 数据库数据源 Myql -->

<DBSource id="Mysql">

<Type>mysql</Type>

<Host>127.0.0.1</Host>

<Port>3306</Port>

<Database>test</Database>

<Username>root</Username>

<!-- AES加密 -->

<Password>A8A47CEBEDE74C314B024D5C5D8F54CB</Password>

<AdditionalInfo/>

</DBSource>

<!-- 数据库数据源 Oracle -->

<DBSource id="Oracle">

<Type>oracle</Type>

<Host>127.0.0.1</Host>

<Port>1521</Port>

<Database>orcl</Database>

<Username>tqz</Username>

<!-- AES加密 -->

<Password>77D8179D71F51C8EDE2C556638667945</Password>

<AdditionalInfo/>

</DBSource>

<!-- 数据库数据源 SqlServer -->

<DBSource id=" MSSQLSERVER ">

<Type>MSSQLSERVER</Type>

<Host>192.168.6.26</Host>

<Port>1433</Port>

<Database>pushdata</Database>

<Username>sa</Username>

<!-- AES加密 -->

<Password>77D8179D71F51C8EDE2C556638667945</Password>

<AdditionalInfo/>

</DBSource>

<!--

文件数据源

可多个，以id唯一标示，用户根据需要自行配置

-->

<FileSource id="FileSource1">

<!--

文件源目录，确保当前用户对该目录下的文件具有读权限

内置3个变量，推送工具会将其替换为实际值

${day} ：表示周期日 格式yyyyMMdd

${day\_ep} ：表示周期日 格式yyyy-MM-dd

${hour} ：表示周期小时 如08

-->

<Directroy>/data2/tqz/UP/operLog/${day}</Directroy>

</FileSource>

<!--

传输服务，采用SCP协议

可多个，以id唯一标示，用户根据需要自行配置

-->

<TransportService id="Data2BI">

<Mode>SCP</Mode>

<!-- 远程主机 -->

<Host>10.120.5.45</Host>

<Port>22</Port>

<Username>qingzheng</Username>

<!-- 验证方式 1-密码 2-公钥 -->

<AuthType>2</AuthType>

<!-- 密码 AES加密 验证方式为1时必填 -->

<Password>873693CC33DDB8CF3C03BD7C0CC30D58</Password>

<!-- 私钥口令 AES加密 生成密钥对时输入的密码短语 验证方式为2时必填 -->

<PrivateKeyPass>873693CC33DDB8CF3C03BD7C0CC30D58</PrivateKeyPass>

<!-- 私钥文件路径 验证方式为2时必填 -->

<PrivateKeyFile>/data2/tqz/.ssh/id\_rsa</PrivateKeyFile>

<!-- 必须确保远程目录存在，并拥有读写权限 -->

<Directory>/data1/MiscTest/qingzheng/DataFile</Directory>

</TransportService>

</Conf>

## 业务配置

业务配置文件AppConf.xml用于配置数据采集推送任务、传输选项等。

数据推送任务, 按数据的来源，可分为2类：

* 数据库类数据推送任务，数据从关系数据库中获取
* 文件类数据推送任务，数据从文本文件(如日志文件)中获取

这些任务用到的数据源、传输服务等在环境配置文件中配置。

### 数据库类推送任务

数据库类推送任务用于从关系数据库中提取数据，并发送到BI环境。

可以配置的项包括：

* 数据源ID -- 从环境配置文件中获取
* 数据提取SQL语句，支持时间变量，用于提取某一周期的数据
* 目标文件名样式 -- 可自由定义，以与现网加载任务适配
* 传输服务 --从环境配置文件中获取
* 传输选项 – 在业务配置文件中配置
* 周期类型 – 支持按天、按小时提取数据

可以配置是否生效，当配置为生效时，推送任务才会被调度执行。

暂支持最多10个DbPush 并发任务。

DbPush配置项说明：

|  |  |  |
| --- | --- | --- |
| 配置项名称 | 含义 | 可能取值 |
| jobName | 任务名称 | 目标数据仓库表名称  日志文件名称 |
| enabled | 是否生效 | true、fasle |
| DBSource | 数据源 | 环境配置文件中的数据源ID |
| ExtractSql | 提取SQL | select \* from ODS\_PUSH\_ROUTERECORD\_DM where updatetime between ‘${startTime}’ and ‘${endTime}’ . 对于从一个数据库取多个独立的表的需求，可以考虑通过”Select ..Join…”操作将多个数据表的记录合并为一张宽表，在接收端根据字段划分将宽表恢复为多个独立的表记录。通过该方式可有效减少任务数（目前工具最多支持10个并行DB任务），提高系统传送效率。 |
| DestFileName | 目标文件名样例 | ${node}\_${jobName}\_${periodTime}.txt |
| TransportService | 传输服务 | 环境配置文件中的传输服务ID |
| TransportConf | 传输选项 | 业务配置文件中的传输选项ID |
| PeriodType | 周期类型 | D – 按天 H – 按小时 |

典型配置实例如下：

<!--

DB类数据推送任务1

jobName: 唯一标示一个Job，用户自行配置

enabled: 是否生效

-->

<DbPush jobName="DbPushJob1" enabled="true">

<!-- 数据源标示，环境配置文件中获取 -->

<DBSource>Mysql</DBSource>

<!--

数据提取SQL语句

内置2个时间变量，推送工具会将其替换为实际值

${startTime}: 周期开始时间，格式yyyy-MM-dd hh:mm:ss

${endTime} ：周期结束时间，格式yyyy-MM-dd hh:mm:ss

-->

<ExtractSql>

<![CDATA[

select \* from ODS\_PUSH\_ROUTERECORD\_DM where updatetime between ‘${startTime}’ and ‘${endTime}’

]]>

</ExtractSql>

<!--

目标文件名样式，用户可自行配置样式

内置3个变量，推送工具会将其替换为实际值

${node} : 节点名称，环境配置文件中获取

${jobName} : 任务名称

${periodTime} ：周期标示，格式yyyyMMddhh，调度系统获取

-->

<DestFileName>${node}\_${jobName}\_${periodTime}.txt</DestFileName>

<!-- 传输服务标示，环境配置文件中获取 -->

<TransportService>Data2BI</TransportService>

<!-- 传输配置标示，应用配置文件中获取-->

<TransportConf>Global</TransportConf>

<!-- 周期类型 D-天 H-小时 -->

<PeriodType>D</PeriodType>

</DbPush>

### 文件类推送任务

文件类推送任务用于从文本文件中提取数据，并发送到BI环境。

* 支持文件个数和大小检查。
* 支持文件记录的过滤转换。

可以配置的项包括：

* 数据源 – 从环境配置文件中获取
* 文件名匹配模式字符串 – 支持文件名通配符
* 文件匹配模式 – 是否需要过滤转换文件内容，
* 过滤转换规则 – 当需要过滤转换文件内容时，支持配置过滤转换规则
* 目标文件名样式
* 支持文件质量检查，包括文件个数和大小的检查，当文件个数或大小不符时支持多次重试
* 传输服务
* 传输选项
* 周期类型

暂支持最多5个FilePush 并发任务。

FilePush配置项说明：

|  |  |  |
| --- | --- | --- |
| 配置项名称 | 含义 | 可能取值 |
| jobName | 任务名称 | 目标数据仓库表名称  日志文件名称 |
| enabled | 是否生效 | true、fasle |
| FileSource | 文件源ID | 环境配置文件中的数据源ID |
| FileMask | 文件名匹配模式 |  |
| FileFetchMode | 文件匹配模式 | 1 - 文件名 匹配到多个文件则合并  2 - 按文件名+记录 支持按记录过滤、转换、提取部分字段、增加缺省字段 |
| FieldSeparator | 源文件分隔符 | 可以为| 、 \001等字符 |
| RecordMask | 记录过滤条件 | 过滤非法值、过滤时间 |
| RecordTransform | 记录转换规则 | 敏感字段替换, 或者删除、增加字段 |
| DestFileName | 文件名样例 |  |
| FileCountThreshold | 最小文件个数 | 文件质量检查条件。对于文件类数据源，提取数据前支持数据质量检查，满足数据质量条件后才启动数据提取。文件质量检查支持重试，以等待源数据文件生成。 |
| FileSizeThreshold | 最小文件大小 |
| ReRetryTimes | 重试次数 |
| WaitTime | 重试等待时间 |
| SourceFilePolicy | 源文件处理测试 | 1 – 删除  2 – 备份  3 – 不处理 |
| TransportService | 传输服务 | 环境配置文件中的传输服务ID |
| TransportConf | 传输选项 | 业务配置文件中的传输选项ID |
| PeriodType | 周期类型 | D – 按天 H – 按小时 |

典型配置实例如下：

* 场景1：将某目录下文件名匹配的文件合并为一个新的数据文件（FileFetchMode =1）
* 对文件类数据提取，支持预先执行文件数据质量检查：即匹配的文件数或文件大小是否满足提取要求。文件检查支持重试，以等待系统生成需要提取数据的源文件。多次重试后仍然无有效源文件，则生成空文件发送。本示例设置源文件个数至少为1，不进行重试。

<!-- 文件类数据推送任务1 -->

<FilePush jobName="FilePushJob1" enabled="false">

<!-- 数据源标示，环境配置文件中获取 -->

<FileSource>FileSource1</FileSource>

<!--

文件名匹配模式，可包含通配符\*?

内置3个时间变量，推送工具会将其替换为实际值

${day} ：表示周期日 格式yyyyMMdd

${day\_ep} ：表示周期日 格式yyyy-MM-dd

${hour} ：表示周期小时 如08

-->

<FileMask>10.120.5.47\_DbPushJob?\_${day}\*.txt</FileMask>

<!--

文件匹配模式

1 - 文件名 匹配到多个文件则合并

2 - 按文件名+记录 支持按记录过滤、转换、提取部分字段、增加缺省字段

-->

<FileFetchMode>1</FileFetchMode>

<!--

目标文件名样式，用户可自行配置样式

${node} : 节点名称，环境配置文件中获取

${jobName} : 任务名称

${periodTime} ：周期标示，格式yyyyMMddhh，调度系统获取

-->

<DestFileName>${node}\_${jobName}\_${periodTime}.txt</DestFileName>

<!--

最少文件个数

当文件个数不足时，可等待，并多次重试

-->

<FileCountThreshold>1</FileCountThreshold>

<!--

最小文件大小

当文件大小不符时，可等待，并多次重试

-->

<FileSizeThreshold>0</FileSizeThreshold>

<!-- 文件个数或大小不符，重试次数 -->

<ReRetryTimes>0</ReRetryTimes>

<!-- 文件个数或大小不符，等待时间，单位分钟 -->

<WaitTime>5</WaitTime>

<!--

数据提取后，源文件处理策略

1：删除

2：备份

3：不处理

-->

<SourceFilePolicy>3</SourceFilePolicy>

<!-- 传输服务标示，环境配置文件中获取 -->

<TransportService>Data2BI</TransportService>

<!-- 传输配置，应用配置文件中获取 -->

<TransportConf>Global</TransportConf>

<!-- 周期类型 D-天 H-小时 -->

<PeriodType>D</PeriodType>

</FilePush>

* 场景2： 文件名及文件记录匹配，将符合文件名及文件记录匹配的数据合并到一个目标文件，并对文件记录进行变换。（FileFetchMode设为2）。

对这种场景，典型配置项包括：

* 提取文件质量检查，参考场景1数据质量检查描述。
* 待过滤记录的列分隔符，本实例设为字符 |, 行分隔符默认是\n.
* 列值过滤条件，支持通过时间变量或常量对列值过滤。过滤字符串是1个符合JAVA语法的关系表达式，当表达式求值为false时，则丢弃该条记录。本实例设置提取周期数据，对第11列值做时间过滤。
* 提取的字段设置。支持对原记录的字段作增加、删除或修改操作。对于修改，支持利用时间变量或常量替换原来的某列数据。示例中设置删除第1列，修改第2列，在2列后插入列值都为null的3列（以配置的列分隔符分开）.

<!-- 文件类数据推送任务2 -->

<FilePush jobName="FilePushJob2" enabled="false">

<!-- 数据源标示，环境配置文件中获取 -->

<FileSource>FileSource1</FileSource>

<!--

文件名匹配模式，可包含通配符\*?

${day} ：表示周期日 格式yyyyMMdd

${day\_ep} ：表示周期日 格式yyyy-MM-dd

${hour} ：表示周期小时 如08

-->

<FileMask>userprofile.operation.log\*</FileMask>

<!--

文件匹配模式

1 - 文件名 匹配到多个文件则合并

2 - 按文件名+记录 支持按记录过滤、转换、提取部分字段、增加缺省字段

-->

<FileFetchMode>2</FileFetchMode>

<!-- 源文件列分隔符 支持 |、\t、\001等字符 -->

<FieldSeparator>|</FieldSeparator>

<!--

记录过滤条件，可多个

示例，如下配置可满足UP按天过滤日志的需求

-->

<RecordMask>

<!-- 列索引，从0开始 -->

<ColumnIndex>11</ColumnIndex>

<!--

列过滤条件

Boolean表达式，必须符合java语法，返回true表示符合过滤条件

用户根据需要可自行实现int、long、Date类型数据的解析

${columnValue}： 列值

${startTime}： 周期开始时间 格式yyyy-MM-dd hh:mm:ss

${endTime} ： 周期结束时间 格式yyyy-MM-dd hh:mm:ss

${day} : 周期日 格式yyyyMMdd

${hour} : 周期小时 格式hh

-->

<ColumnFilter>

<![CDATA[

new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").parse("${columnValue}").compareTo(new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").parse("${startTime}")) >= 0

&&

new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").parse("${columnValue}").compareTo(new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").parse("${endTime}")) <= 0

]]>

</ColumnFilter>

<!-- 满足UP日志提取部分字段需求 -->

<RecordTransform>

<ColumnIndex>1</ColumnIndex>

<!--

1 – 删除

2 – 转换

3 – 添加缺省字段

-->

<Mode>1</Mode>

</RecordTransform>

<!-- 满足UP日志特定字段敏感化处理需求 -->

<RecordTransform>

<ColumnIndex>2</ColumnIndex>

<Mode>2</Mode>

<Format>\*</Format>

</RecordTransform>

<!-- 满足UP日志增加部分缺省字段需求 -->

<RecordTransform>

<ColumnIndex>3</ColumnIndex>

<Mode>3</Mode>

<Format>null|null|null</Format>

</RecordTransform>

</RecordMask>

<!--

目标文件名样式，用户可自行配置样式

${node} : 节点名称，环境配置文件中获取

${jobName} : 任务名称

${periodTime} ：周期标示，格式yyyyMMddhh，调度系统获取

-->

<DestFileName>${node}\_${jobName}\_${periodTime}.txt</DestFileName>

<!-- 最少文件个数 -->

<FileCountThreshold>1</FileCountThreshold>

<!-- 最小文件大小 -->

<FileSizeThreshold>0</FileSizeThreshold>

<!-- 文件个数或大小不符，重试次数 -->

<ReRetryTimes>0</ReRetryTimes>

<!-- 文件个数或大小不符，等待时间，单位分钟 -->

<WaitTime>5</WaitTime>

<!--

数据提取后，源文件处理策略

1：删除

2：备份

3：不处理

-->

<SourceFilePolicy>3</SourceFilePolicy>

<!-- 传输服务标示，环境配置文件中获取 -->

<TransportService>Data2BI</TransportService>

<!-- 传输配置，应用配置文件中获取 -->

<TransportConf>Global</TransportConf>

<!-- 周期类型 D-天 H-小时 -->

<PeriodType>D</PeriodType>

</FilePush>

### 目标文件格式配置

支持提取到的目标数据文件的行列格式配置，可以配置列分隔符、行分隔符。

默认列分隔符为竖线|，行分隔符为换行符 \n。

<!-- 目标文件格式 -->

<DestFileFormat>

<!-- 列分隔符 支持 |、\t、\001等字符 -->

<FieldSeparator>|</FieldSeparator>

<LineSeparator>\n</LineSeparator>

</DestFileFormat>

### 传输选项配置

传输选项对传输过程进行控制，可以设置是否压缩加密、大文件分割、失败重试等。

* 过大的文件无法执行压缩，发送周期过长也增加了发送失败的几率，故实现过程中先把大文件分割成若干个小文件，然后分别对小文件执行压缩传输，在接收端再对小文件执行合并。
* 建议1个文件在1分钟内发送完毕，参考现网2M的带宽及压缩比率，大文件分割大小在600M左右为宜。
* 可设置多个传输选项，分别对推送任务进行个性化传输控制。

TransportConf 配置项说明：

|  |  |  |
| --- | --- | --- |
| 配置项名称 | 含义 | 可能取值 |
| IsCompressAndEncrypt | 是否压缩 | true，false |
| Password | 压缩加密密码 |  |
| SplitSize | 大文件分割大小 |  |
| ReSendTimes | 文件发送失败，重试次数 |  |
| WaitTime | 文件发送失败，等待时间，单位为分钟 |  |
| SourceFilePolicy | 发送成功后，目标文件处理策略 |  |

典型配置实例如下：

<!--

传输配置

可多个，以id唯一标示，用户根据需要自行配置

-->

<TransportConf id="Global">

<!-- 是否压缩加密 -->

<IsCompressAndEncrypt>true</IsCompressAndEncrypt>

<!-- 加密密码，AES128加密，当压缩加密时必填 -->

<Password>7BE216A80761D3875576BFDD422A3D65</Password>

<!-- 大文件分割大小，单位M -->

<SplitSize>600</SplitSize>

<!-- 文件发送失败，重试次数 -->

<ReSendTimes>3</ReSendTimes>

<!-- 文件发送失败，等待时间，单位分钟 -->

<WaitTime>1</WaitTime>

<!--

发送成功后，目标文件处理策略

1：删除

2：备份，备份目录: xxx/工具包/backup

-->

<SourceFilePolicy>2</SourceFilePolicy>

</TransportConf>

## 日志配置

推送工具采用log4j作为日志组件，可以设置日志级别、输出目的，对日志输出进行控制。

通常，调试阶段，日志级别可设置为DEBUG级别，添加console输出。

**在线运行时需将日志级别设置为INFO，去掉console输出！**

#日志级别

log4j.rootLogger=DEBUG, console, file

# 运行

## 自动调度

推送工具需要传入2个调度参数：

* 周期时间 – 格式：yyyyMMddhh，表示提取某1周期的时间
* 周期类型 – D –天 H小时，支持按天、按小时提取数据。

推送工具不提供调度功能，业务可采用cron或TCC调度推送工具以完成数据推送任务。

为方便cron调度，本工具提供配套度脚本*DataPush4cron.sh*，脚本支持设置周期类型、相对当前时间的周期偏移量。通常，推送任务按天定时执行，提取前一天的数据。故周期类型可设置为D，偏移量可设置为-1。

如有按小时执行的任务，需单独调度。

调度脚本*DataPush4cron.sh*内容：

#!/bin/bash

# 周期类型：D天 H小时

PERIOD\_TYPE=D

# 当前时间的周期偏离量，负数表示向前偏移，正数表示向后偏移，单位取周期类型。

OFFSET=-1

PERIOD\_TIME=""

if [ $PERIOD\_TYPE = "D" ]; then

PERIOD\_TIME=`date +%Y%m%d%H --date="$OFFSET day"`

elif [ $PERIOD\_TYPE = "H" ]; then

PERIOD\_TIME=`date +%Y%m%d%H --date="$OFFSET hour"`

else

echo "invalid PERIOD\_TYPE: $PERIOD\_TYPE"

exit -1

fi

# echo $PERIOD\_TIME

ROOT\_PATH=`dirname $0`

JOB\_PATH=$ROOT\_PATH/DataPush\_0.1/DataPush/DataPush\_run.sh

sh $JOB\_PATH --context\_param periodTime=$PERIOD\_TIME --context\_param periodType=$PERIOD\_TYPE

执行crontab –e 命令添加调度

***cron调度举例：***

0 1 \* \* \* /home/hibi/DataPush/DataPush4cron.sh 1>>/home/hibi/DataPush/log/localhost.log 2>&1

注意：cron不识别用户的环境变量，故需要大家在DataPush/DataPush\_0.1/DataPush/DataPush\_run.sh中添加**java全路径**。

## 手动执行

举例：/home/hibi/DataPush/DataPush\_0.1/DataPush/DataPush\_run.sh --context\_param periodTime=2012090708 --context\_param periodType=D --context\_param jobName=jobName1

根据需要设置参数periodTime、periodType、jobName的取值。jobName1指只执行某一任务，如果不指定该参数，则配置文件中的所有任务都将被调度。

## 日常维护

当传输选项的SourceFilePolicy配置为备份时，工具包的backup目录下的备份文件会越来越多，可能会占用过多磁盘空间。因此有必要及时清理备份文件。

提供办法：

1. 可以将SourceFilePolicy配置为删除，则提取到的目标文件发送成功后不备份，被立即删除。
2. 也可以写个删除脚本，定时删除备份文件。比如每天执行，删除3天前的备份文件。

提示：提取到的数据文件，发送到BI环境后，通常会很快被加载到数据仓库，因此在业务侧，数据文件并不需要长时间的备份。

# 附录

## SSH密钥认证

1. 本地生成密钥对

执行*ssh-keygen -t rsa*命令生成密钥对，其中id\_rsa为私钥，id\_rsa.pub为公钥。

可以设置密钥保存路径，默认在用户家目录下。

可以设置私钥密码，如设置私钥密码，则私钥密码必须大于4个字符，需要用户记住这个私钥密码；如不设私钥密码，可以直接登录远程主机，但不够安全。

以下为命令实例：

qingzheng@TestGateway:~> ssh-keygen -t rsa

Generating public/private rsa key pair.

Enter file in which to save the key (/home/MiscTest/qingzheng//.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/MiscTest/qingzheng//.ssh/id\_rsa.

Your public key has been saved in /home/MiscTest/qingzheng//.ssh/id\_rsa.pub.

The key fingerprint is:

43:26:2b:1e:75:7f:1e:80:cf:e3:5e:de:5f:eb:0c:e5 qingzheng@TestGateway

The key's randomart image is:

+--[ RSA 2048]----+

| |

| . |

| o = . |

| . \* + . |

| o . S = o . |

| . o o + .o |

| . . o. E.|

| . o .o o|

| . . o=.|

+-----------------+

1. 将公钥copy到远程主机，可以通过scp命令远程拷贝，实例：

scp id\_rsa.pub root@192.168.1.39:/root/

1. 将公钥添加到远程主机的authorized\_keys文件中

authorized\_keys文件通常在登陆用户家目录的.ssh目录下

登陆远程主机，执行以下命令完成公钥添加：

cat id\_rsa.pub >> .ssh/authorized\_keys

1. 验证密钥登陆

ssh [$user@192.168.1.39](mailto:$user@192.168.1.39)

如果没有设置私钥密码可以直接登录；如果设置了密码，请输入私钥密码登陆。

1. 推送工具中配置密钥登陆方式

需要配置私钥密码和私钥文件路径

## FAQ

暂无