

大数据开发套件

常见问题

常见问题

Q：大数据开发平台上的项目怎么绑原来已有的ODPS project？

A：主账号在新建大数据开发套件项目的时候，选择已有ODPS Project对应的计费方式，项目名称填写已有ODPS Project Name，并点击确定，即可添加已有ODPS Project 到大数据开发套件中。

Q：我公司已经有创建组织了，怎么把我加为成员？

A：联系组织管理员索要子账号，联系对应项目管理员将该子账号添加至对应项目中。添加成功子账号登录即可。相关子账号登录问题，可以参考《首次使用》。

Q：如何授权？

A：将子账号添加至项目中，授予不同角色身份，以获取相应的操作权限。

注释：账号进入租户后默认没有任何权限（创建者项目除外），账号被添加至某一个项目中后，会根据相应的授权角色获取相应的权限。

Q：修改了主账号的登录名，用修改后的登录名将子账号拉进A项目中，子账号使用A项目时，报错：“非法的子账号，不是组织主账号的子账号”。

A：使用主账号在<https://data.aliyun.com/console/member> 这个页面，点击一下刷新用户即可。

数据集成常见问题

整库迁移目前只提供MySQL（包括rds里的MySQL）同步到MaxCompute，可以从已经添加好的MySQL数据源里进入整库迁移界面，如下图所示：

下面只针对整库迁移里高级设置的数据类型的介绍，如下图所示：

整库迁移源端MySQL支持的数据类型：tinyint、smallint、mediumint、int、bigint、varchar、char、tinytext、text、mediumtext、longtext、year、float、double、decimal、date、datetime、timestamp、time、bool
 目标端MaxCompute支持的数据源类型：bigint、string、bouble、datetime、boolean
 上面MySQL支持的数据类型支持都与MaxCompute数据源类型之间的转换。

注意：mysql里的bit，如果是bit(2)以上，则目前不支持与bigint、string、bouble、datetime、boolean这些类型转换；如果是bit(1)，则会被转换成boolean

数据背景：表是按天分的（如orders_20170310,orders_20170311,orders_20170312），每天一个表，表结构一模一样。

需求：创建数据同步任务将这些表导入到MaxCompute中，希望只需要创建一个同步任务，表名自定义，实现每天凌晨自动从源数据库读取昨天的表数据（如，今天是2017年3月15日，自动从源数据库中读取orders_20170314的表的数据导入，以此类推）。

实现方式：

步骤一：在大数据开发套件->数据集成通过向导模式创建数据同步任务，配置时数据来源表先选一个表名如orders_20170310。按正常配置流程配置好同步任务并保存好。

步骤二：转换脚本，把向导模式转成脚本模式。



步骤三：在脚本模式中，将来源表的表名改用变量如orders_`\${tablename}`。然后任务的参数配置中，给变量tablename赋值。因“数据背景”中表名是按天区分，而需求是每天读取昨天的表，所以赋值为`\${yyyyymmdd-1}`。

或者，来源表的表名改用变量时，直接写成orders_`\${bdp.system.bizdate}`

步骤四：把整个任务其他配置项配置好后，保存提交，再进行后续的其他操作。

数据集成的同步任务配置编码格式后，可能出现以下两种情况：

同步失败且产生脏数据

错误问题：

数据集成任务失败，且因编码问题产生脏数据。

报错日志如下所示：

```
016-11-18 14:50:50.766 [13350975-0-0-writer] ERROR StdoutPluginCollector - 脏数据:<br>
{"exception":"Incorrect string value: '\\xF0\\x9F\\x98\\x82\\xE8\\xA2...' for column 'introduction' at row 1", "record":{"byteSize":8, "index":0, "rawData":9642, "type":"LONG"}, {"byteSize":33, "index":1, "rawData":"优衣库兼职出来的女汉子，扛得了箱子，招待好顾客![1](http://docs-aliyun-cn-hangzhou.oss.aliyun-inc.com/assets/pic/56134/cn_zh/1498728641169/%E5%9B%BE%E7%89%877.png)被自己感动cry", "type":"STRING"}, {"byteSize":8, "index":4, "rawData":0, "type":"LONG"}], "type":"writer"}
2016-11-18 14:50:51.265 [13350975-0-0-writer] WARN CommonRdbmsWriter$Task - 回滚此次写入, 采用每次写入一行方式提交. 因为:java.sql.BatchUpdateException: Incorrect string value: '\\xF0\\x9F\\x88\\xB6\\xEF\\xB8...' for column 'introduction' at row 1
```

错误原因：

用户要对数据库做相应的编码格式设置或在添加数据源时相应的将编码设置成utf8mb4，这样数据集成支持同步表情符。

解决方法：

添加 jdbc 格式的数据源时，需要修改 utf8mb4 的设置，例如：
jdbc:mysql://xxx.x.x.x:3306/database?com.mysql.jdbc.faultInjection.serverCharsetIndex=45，这样在数据源设置表情符可以同步成功；

将数据源编码格式改成 utf8mb4。例如：在 rds 管控台修改 rds 的数据库编码格式。

同步成功但数据乱码

错误问题：

虽然数据同步任务成功了，但是数据发生乱码。

错误原因：

发生乱码的原因有以下三种：

```
源端是本来就乱码；
数据库和客户端编码不一样；
浏览器编码不一样，导致预览失败或乱码。
```

解决方法：

原始数据乱码时，需用户处理好原始数据后再做同步任务；
数据库和客户端编码不一样，需先将编码格式进行修改；
浏览器编码不一致，需先将编码格式进行统一，然后进行数据预览。

如果问题仍未解决，请联系 售后技术支持。

Q：如何创建 MaxCompute 表？

A：大数据平台上有两种方式建表：

在数据开发模块通过脚本文件或者工作流任务执行建表语句，详情请看 [快速开始-创建表](#)。以这种方式建表，owner 为项目管理中项目计算引擎设置的 'MaxCompute 访问身份' 账号，无类目管理。

是在数据管理模块可视化方式建表，详情请看 [数据管理-创建表](#)。以这种方式建表，owner 为当前登录账号，有类目管理。

Q：怎么把数据导入到 MaxCompute 上？

A：数据导入到 MaxCompute 基本归类为 2 种情况：

1) 本地数据文件一次性导入到 MaxCompute 的某张表；

解决方案：直接在数据开发面板右边工作区域右上角“新建-导入本地数据”，详细请参考文档 [数据导入](#)。

2) 离线数据库单表导入。

解决方案：在“数据集成”界面，点击左侧导航栏 同步任务，创建‘数据同步’任务。具体可以参考 [创建数据同步任务](#)。

Q：RDS 数据源连通测试不通过怎么办？

A:目前 RDS 数据源仅支持杭州地域的 RDS，北京地域暂时不支持。另外当杭州地域的 RDS 也遇到数据源测试不连通的时候，需要到自己 RDS 上添加数据同步机器 IP 白名单：

10.152.69.0/24,10.153.136.0/24,10.143.32.0/24,120.27.160.26,10.46.67.156,120.27.160.81,10.46.64.81,121.43.110.160,10.117.39.238,121.43.112.137,10.117.28.203,118.178.84.74,10.27.63.41,118.178.56.228,10.27.63.60,118.178.59.233,10.27.63.38,118.178.142.154,10.27.63.15,100.64.0.0/8

注意：若使用自定义资源组调度 RDS 的数据同步任务，必须把自定义资源组的机器 ip 也加到 RDS 的白名单中。

Q：调度参数怎么用？

A：目前系统参数有两个，这两个系统参数只要代码里用到，不需要做额外的赋值动作，节点自动调度的时候就会自动替换。系统参数分别为：

`{bdp.system.bizdate}`格式 `yyyymmdd`，日常调度实例定时时间的前一天（年月日）。节点每天自动调度实例定时时间年月日减 1 天。

`{bdp.system.cyctime}`格式 `yyyymmddhh24miss`，日常调度实例定时时间（年月日时分秒）。
`yyyy` 表示 4 位数年份，`mm` 表示 2 位数月份，`dd` 表示 2 位数天，`hh24` 表示 24 小时制的时，`mi` 表示 2 位数分钟，`ss` 表示 2 位数秒。

使用方式：节点代码里编辑如 `pt={bdp.system.bizdate}`，然后节点在调度上运行即可自动替换值。

若节点每天调度时代码里有变量需要赋当天减 2 的日期，即业务日期减 1 天，格式为 `yyyymmdd`，需要系统参数和 MaxCompute 内建函数一起使用，如下：

节点代码 `dt=to_char(dateadd(TO_DATE('{bdp.system.bizdate}' , '{yyyymmdd}'), -1, 'dd'), '{yyyymmdd}')`

MaxCompute 内建日期函数具体可查看帮助文档：大数据计算服务 MaxCompute > SQL > **内建函数** 定位到“日期函数”。

Q：本地文件数据上传功能有没有文件大小限制？

A：目前支持.csv/.txt 文件上传，一次性上传文件大小上限为 10M。

Q：我在“数据开发”模块通过脚本写建表语句建成功建表，但是为什么在“数据管理”模块“数据表管理-我拥有的表”列表里没看到？**

A：遇到此问题，首先要查看当前项目的“项目管理-MaxCompute 基本配置”，MaxCompute 访问身份配置应该是选择了“系统账号”，通过数据开发脚本提交的建表语句都是用该账号创建的表即表 owner 为该账号，而“数据管理-我拥有的表”是指当前登录账号为 owner 的表。

Q：一个工作流里的节点可以不彼此依赖吗？

A：

- 一个工作流里没有业务关系的节点可以不用彼此依赖；
- 一个工作流里只能有一个根节点，即没有父节点的节点；
- 一个工作流里某个节点可以依赖多个节点，也可以被多个节点依赖；

- 一个工作流里可以有多个节点没有被其他节点依赖。

Q：如何将 RDS 里的数据表链接到 Data IDE 中？

A：大数据开发套件（Data IDE）只能查看 MaxCompute 的数据，RDS 的数据表需要同步到 MaxCompute 的表才可以在 Data IDE 中操作。数据同步可以参考：
https://help.aliyun.com/document_detail/30269.html

Q：数加平台能不能从 RDS 的只读实例同步数据？

A：支持从只读 RDS 读出数据，然后导到其他地方。实际上，为了减少对业务库的压力，我们建议做 ETL 的时候，从只读库上导。具体数据同步配置可参考：https://help.aliyun.com/document_detail/30269.html

Q:什么是配置跨周期依赖？

A：跨周期依赖分为以下几种：

自依赖。

场景举例：用户配置小时任务所统计，拿上一个小时的统计结果汇总新增数据，产出当前这个小时的统计数据。用户需要针对当前小时任务配置自依赖，保证每个小时触发执行的时候上一个小时已经执行完了。

等待下游任务的上一周期执行完成。

场景举例：任务 A 用于汇总存量数据和新增数据，任务 B 用于清洗数据并将结果数据写入存量数据，任务 B 是依赖任务 A 的。用户需要将任务 A 跨周期依赖配置为“等待下游任务的上一周期执行完成”，以此保证自己执行的时候，上一个周期的 B 已经执行，存量数据已经是最新的。

等待自定义任务的上一周期结束。

场景举例同上一个类似，只不过任务 A、任务 B 之间没有相互依赖关系。在此不再赘述。

Q：跨周期依赖的场景有哪些？

A：* 天周期任务依赖小时周期任务。

场景举例：用户配置了一个天任务，依赖小时任务，天任务对小时任务做汇总，需要每天 24 个小时任务全都跑完之后才能执行。用户需要针对小周期配置自依赖，保证上游的小时任务全都执行完成。

Q:同事离职后，修改任务流的责任人，却发布失败，提示节点责任人不在工作流中。

A：1、先在运维中心修改责任人，修改时展开工作流列表，全选工作流和节点，然后修改责任人。如下图所示：



2、修改责任人后，在数据开发中找到该工作流，查看责任人是否被修改，确定无误后在工作流页面随意移动一下节点(修改一下工作流)，然后提交，重新打包、发布。如下图所示：



Q： workflow节点出错怎么看日志？

A：进入“运维中心-任务运维-运维”，过滤条件：输入任务名称、业务日期选“接到的节点实例出错提醒的业务日期”，查询，结果列表选中任务实例，右边 DAG 图里，鼠标对实例右键点击‘查看节点运行日志’。

Q：刚提交好了一个工作流，设置为周期调度，定时时间已经到了，怎么还不见跑起来？

A：当天新增成功的工作流/节点，且在当天 23：30 前提交/发布成功，需要第二天才会自动调度。当天可以通过手动测试或补数据调度。

Q：创建了周期调度工作流，且昨天就 23：30 前就发布成功了

，今天运维中心里却是未运行状态？

A：以下几种原因都会导致 workflow 未运行，注意逐一排查：

1. workflow 属性-调度属性-启动调度选项没有勾上。
2. workflow 虽然勾选 ‘启动调度’ 但是今天的日期不在设置的 ‘生效时间’ 范围内。
3. workflow 设置的定时调度时间还没到。
4. workflow 设置有上游依赖，上游 workflow 未全部运行成功。
5. workflow 设置有跨周期依赖，上一周期 workflow 未运行成功。

说明：节点任务排查方式与 workflow 相同。

Q：workflow 前段时间每天都正常调度，今天突然没有了，workflow 实例都没生成。

A：请注意查看 workflow 是否已经删除，或者 workflow 属性的生效时间已经过了，若生效时间已过可以修改。

Q:数据同步任务等待资源时间过长，一直等待资源？

A：如果任务使用自定义资源组，那么：

1. ECS 上已经跑了太多的任务，导致任务排队；
2. ECS 节点无法正常工作，可以到 组织管理->调度资源->资源服务器 看下 ECS 资源是否正常，也可以考虑重启一下 ECS 上的 agent，命令是:su - admin/home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart

如果任务使用默认资源组，那么就是资源组并发已满，需要等待或者可以考虑使用自定义资源组。

Q：数据同步任务报错：获取作业配置信息失败 :/home/admin/shell_datax/T3_0028378354 - File '/home/admin/shell_datax/T3_0028378354' does not exist

A：这个错误一般是通过自定义资源组执行 shell 调 datax 的任务，可以在自定义机器上 su - admin 切换到 admin 账号后，执行下 datax 任务调试下，常见原因：

- 权限问题（datax 的任务是用 admin 账号执行的，如果用 root 配置了环境，可能有一些操作 admin 账号会没权限）。
- datax 配置的时候就有问题，无法正常执行。

如果还是无法解决请提交工单寻求帮助！

Q：整个工作流中间的一个节点出错，能否跳过先执行其他的节点？

A：如果节点没运行成功，下游节点是不能正常运行。若节点之间业务逻辑没有依赖关系，则应该取消其直接的调度依赖；若仅仅是本次执行可以不依赖出错节点的执行结果，则可以对出错的节点实例操作右键“置成功并恢复调度”这样可以先跳过出错节点调度下游节点。

Q：为什么按照流程部署完毕了 ECS，但界面还是显示“终止”？

A：

- 如果是经典网络，请确认您注册使用的机器名称是真实的机器名,请使用命令到ecs上执行：`hostname`，返回的结果即为机器名称；不支持自定义名称；
- 如果是专有网络，请确认使用的是界面提供的命令查询到的 `uuid`；
- 请确认是否有修改过 `ecs` 的主机名称，注意这里不是指实例名。如果修改过名称，请到 `ecs` 上查询 `/etc/hosts` 中是否有绑定正确；
- 如果注册信息都是正确的，而且也是按照“执行初始化”的命令复制粘贴部署的，但还是“终止”，请将 `ecs` 上的文件 `/home/admin/alisatasknode/logs/alisatasknode.log` 这份文件转为 `txt` 格式，提工单反馈给我们；

Q：配置服务器的时候，在操作“执行初始化”的时候失败？

A：目前只能支持 `centos5`、`centos6`、`centos7` 以及 `aliyunos`。如果您的 `ecs` 不是上述的 `os`，系统暂时不支持，会出现上述操作。

Q：为什么添加了调度资源，也安装 ECS 成功了，但是 shell 任务还是执行失败，报错“exec target was null”？

A：执行 `shell` 任务，需要将任务的运行资源组修改为自定义资源组；

如何新建自定义资源组请前往《项目管理》文档中查看。如果已经拥有自定义资源组，请前往运维中心修改任务执行的资源组。

Q：为什么我注册了 ECS 服务器，经常会出现自动的“暂停服务”的状态？

A：在注册 `ECS` 的时候，部署了一个 `agent`，`agent` 会有一个监听动作，发现当前 `ECS` 的内存、`cpu`、`load` 较高的时候，不适合继续接收新的任务执行，会主动将 `agent` 服务暂停。待机器指标下降后会自动置为“正常”

” 状态；

Q：为什么安装 ECS 的时候，部署完成了，但是启动失败？

A：请检查您的 ECS 系统是否有 jdk1.6 以上的环境；如果您是 centos5.x 的操作系统需要您自己安装 jdk 的版本以及 python2.6.5 以上的版本；

Q：如何修改自定义 ECS 的任务并发数？

A：在配置服务器的界面中，有个最大槽位数：说明：槽位数是定义一台机器并发任务的一个量化单位，一个 shell 任务或者 MaxCompute sql 任务占用一个槽位，一个同步任务可能会占用 2-20 个不等的槽位。可以修改该值增加您 ECS 并发的大小，目前默认是 40 个槽位；

Q：为什么我的调度资源偶尔会出现“暂停服务”的状态？

A：由于在服务器上安装的 agent 会自动监控当前 ECS 的 cpu、mem、load 的使用率，监控的频率是每分钟监控 3 次，当发现使用率超过 80% 就会将 agent 的服务挂起，也就是在界面看到的“暂停服务”；

注意：该功能并不会影响正在执行的任务，但是不会在接收新的任务，待负载等指标下降后，会自动将服务器状态置为“正常”，服务器恢复正常工作。

Q：为什么跑在自定义调度资源上的任务执行成功/失败，但是日志无法获取？

A：

- 如果服务器是经典网络的，那么请确认内网入的 8000 端口是否开通了；
- 在调度资源中检查当前服务器的状态，是否“正常”；
- 如果上两步骤都是正常的，那么尝试重启下 agent 的服务，重启命令：`sudo su - admin /home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart`

Q：为什么同步任务执行成功了，但是日志中却有500错误的返回？

A：任务成功了，只是日志读取失败了，是因为 ECS 服务 agent 异常，需要重启下 agent，重启命令：`sudo su - admin /home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart`重启完毕后，可以重跑下同步任务，在检查日志是否完整；

Q：为什么任务经常处于“等待资源”？

A:

1. 如果任务是跑在**默认资源组**上：我们默认给每个租户50个槽位的并发数，可能您当前的任务已经占用超过了50个槽位，所以正在等待资源。
2. 如果任务是跑在**自定义资源组**上：那么可以到调度资源管理界面选择对应调度资源，点击服务器管理，查看服务调度资源的服务器状态。

- **停止状态**：请登录 ECS 执行命令 `sudo su - admin /home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart` 恢复 agent 服务；
- **暂停服务状态**：那么是很可能因为 ECS 上跑的任务占用了较高的 cpu 或者内存，导致了挂起。建议如果任务并发量过高，建议将 ECS 的内存 cpu 扩容下；
- **正常状态**：需要检查下当前是否任务数已经达到了 ECS 服务器的并发数上限。可以到运维中心中检查，当前调度资源下有哪些正在执行的任务。

Q：任务没有运行日志？

A：无法获取运行日志（一般出现在使用自定义调度资源的情况下），下面就说一下两种处理方案。

1. 自定义调度资源的机器在初始化时没有对我们读取日志的机器开放白名单以及8000端口，详细操作请参考新增调度资源，详细请参考“[开通 8000 端口，以便读取日志](#)”。
2. agent是我们去连接您机器的服务，读取日志失败可能是agent服务挂了，可以使用如下命令重启：
`sudo su - admin /home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart`

我们的机器只会拉取一次日志数据，如果第一次拉取日志的时候失败了，那么这个任务的日志就无法获取了，只能再次运行该任务后，才会重新拉取一次。

Q：新建Data IDE项目空间时采用绑定已有的ODPS project的方式，新建成功后进入至Data IDE数据管理模块发现ODPS project中原已有的表都没有在其列表中展示。

A：遇到此问题，属于正常范畴。采用这种方式构建Data IDE项目空间，ODPS project中已有的表需要T+1的时间才能够在数据管理模块列表中显示。也就是说需要到第二天的8点-10点之后才能够显示。

Q：在数据开发里可以执行建表语句成功建表，但是在数据管理创建表却说没权限？

A：遇到此问题，首先要查看当前项目的“项目管理-项目属性-技术引擎配置-ODPS访问身份”，这个配置应该是选择了“计算引擎指定账号”，通过数据开发脚本提交的建表语句都是用该账号创建的表，该账号拥有一般都会有权限；而数据管理建表用的是当前登录账号，该账号需要权限必须找项目owner给开通。详细请文档中心 [数据开发套件-数据管理手册-创建表章节的权限准备内容](#)。

Q:表的真实存储信息和表详情页里看到的存储信息不一致。

A:数据管理模块表存储信息是离线计算得出的，默认是一天同步一次这个信息，非实时同步所以与真实信息会有些出入。

Q：我的任务需要去查询别人的表，但是没权限，可以到哪里申请？

A：导航至 " 数据管理模块 > 查找数据 > 申请数据权 " ，如下图所示：



提交申请后，等待表所属项目的管理员审批，审批入口在 " 数据管理模块 > 权限管理 > 待我审批 "。

Q：被添加为项目成员后，就会拥有该项目所有表查询权限吗？

A：请管理员授予项目成员开发角色权限，即可拥有查询该项目下所有表的权限。

Q：数据管理模块申请权限了，但是表删除重建后权限就丢失了？

A：数据管理模块权限申请走的是 MaxCompute 的 ACL 授权方式，表删除会把该表相关的 ACL 授权都清空，重建不会恢复。目前这个现象流程上还没有更好的解决方式，只能项目管理员命令行通过 package 方式授权解决。详情请参见：[MaxCompute 安全文档](#)。

Q：为何A项目的开发角色，可以操作B项目下的表呢？

A：使用主账号看看项目管理下的Maxcompute配置，Maxcompute访问身份设置的是不是系统账号。

Maxcompute访问身份：是指SQL执行的时候，是使用系统账号的AK去执行，还是使用个人账号的AK去执行。

用户常见问题反馈

问题描述

如何跨项目配置大数据开发套件的任务依赖。

处理办法

用户可以在脚本开发页面配置任务依赖的时候，设置任务依赖为其他项目的依赖：

用户可以先输入需要依赖的任务所在的项目空间，然后选择上游任务。如果任务比较多，也可以输入关键字进行过滤。

注意：目前只支持同一个组织下的不同项目才可以跨项目依赖。

调度属性 ▾

调度状态: 暂停

生效日期: 1970-01-01 至 2115-12-19

*调度周期: 天

*具体时间: 00 时 00 分

依赖属性 ▾

自动推荐

所属项目: 请输入项目

上游任务: 请输入关键字查询上游任务

项目名称	任务名称	责任人	操作
没有依赖上游任务			

跨周期依赖 ▾

- 不依赖上一调度周期
- 自依赖，等待上一调度周期结束，才能继续运行
- 等待下游任务的上一周期结束，才能继续运行
- 等待自定义任务的上一周期结束，才能继续运行

调度配置

如果问题还未能解决，请联系售后技术支持。

问题描述

用户配置的同步任务运行失败，伴有报错日志：

- ErrorMessage:Code:[OdpsWriter-10], Description:[提交 block 到 ODPS 目的表失败]. - ODPS 目的表在提交 block:[0] 时失败, uploadId=[XXXXX]. 请联系 ODPS 管理员处理. -
- ErrorCode=DefaultClusterNotLatestVersion, ErrorMessage=The specified table is not the latest version on default cluster and can't upload data because the data is being replicated at this time. Please initiate another upload later.

问题原因

该错误通常是由于数据同步任务涉及的MaxCompute表，在同步的过程中发生了变化导致的。

处理办法

用户可以等前面的数据导入任务结束后对这个任务再进行重试。另外因为MaxCompute里没有事物的概念，需从业务上做好任务依赖。用户可以通过大数据开发套件里的任务的依赖关系功能，保证上下游任务有序进行。

如果问题还未能解决，请联系售后技术支持。

如果没有配置好shell环境，请参考此链接的文档：[Q：怎么顺利执行Shell/MR类型任务](#)

配置完成后，在shell节点填写代码。可以参考下面的demo代码

- 代码说明

- a. 生成临时datax配置文件（您只需要修改配置文件内容即可，其他不用更改。参

考文档：<https://github.com/alibaba/DataX/wiki/Quick-Start>

b. 读取调度参数，分别为\$1, \$2

- `{bdp.system.bizdate}`, `{bdp.system.cyctime}` 不需要配置，具体含义参考：系统参数

c. 执行datax任务，进行数据同步

d. 删除临时文件

e. 判断任务成功失败，进行返回。0代表成功

```
#!/bin/bash
#*****#
##author:yixiao streamreader->streamwriter
##create time:2016-03-28 10:00:00
#*****#

shell_datax_home='/home/admin/shell_datax'
mkdir -p ${shell_datax_home}
shell_datax_config=${shell_datax_home}/${ALISA_TASK_ID}
echo ""
{
"job": {
"setting": {
"speed": {
"channel": 1
},
"errorLimit": {
"record": 0,
"percentage": 0.02
}
},
"content": [
{
"reader": {
"name": "streamreader",
"parameter": {
"column": [
{
"value": "${bdp.system.bizdate}",
"type": "string"
},
{
"value": "${bdp.system.cyctime}",
"type": "string"
},
{
"value": "${params1}_${params2}",
"type": "string"
},
{
"value": 19890427,
"type": "long"
},
{
"value": "1989-06-04 00:00:00",
```

```
"type": "date"
},
{
  "value": true,
  "type": "bool"
},
{
  "value": "test",
  "type": "bytes"
}
],
"sliceRecordCount": 10
}
},
"writer": {
  "name": "streamwriter",
  "parameter": {
    "print": true,
    "encoding": "UTF-8"
  }
}
}
]
}
}
""" > ${shell_datax_config}

params1=$1
params2=$2
datax_params='-p "-Dparams1=${params1} -Dparams2=${params2}"'

echo "`date '+%Y-%m-%d %T` shell datax config: ${shell_datax_config}"
echo "`date '+%Y-%m-%d %T` shell datax params: -p \"-Dparams1=${params1} -Dparams2=${params2}\""
/home/admin/datax3/bin/datax.py ${shell_datax_config} -p "-Dparams1=${params1} -Dparams2=${params2}"
shell_datax_run_result=$?

rm ${shell_datax_config}

if [ ${shell_datax_run_result} -ne 0 ]
then
echo "`date '+%Y-%m-%d %T` shell datax ended failed :("
exit -1
fi
echo "`date '+%Y-%m-%d %T` shell datax ended success ~ "
```

3. 调度系统配置参数

The image shows a code editor interface with a file named 'create_table_ddl'. The code includes a JSON configuration for a 'streamwriter' and a shell script. The shell script defines two parameters, 'params1' and 'params2', and uses them in a 'datax_params' variable. A red box highlights the parameter assignment lines: 'params1=\$1' and 'params2=\$2'. To the right, a panel titled '自定义参数配置' (Custom Parameter Configuration) contains a text input field with the value '123 abc', also highlighted by a red box.

```
59     },
60     "writer": {
61       "name": "streamwriter",
62       "parameter": {
63         "print": true,
64         "encoding": "UTF-8"
65       }
66     }
67   }
68 }
69 }
70 }
71 ''' > ${shell_datax_config}
72
73
74 params1=$1
75 params2=$2
76 datax_params="-p -Dparams1=${params1} -Dparams2=${params2}"
77
78 echo "date +%Y-%m-%d %T" shell datax config: ${shell_datax_config}"
```