# Dship测试用例

## 简单介绍

### 几个概念

* 上传：支持文件和目录（只一级目录）的上传，只支持数据上传到一个table或一个partition。
* 下载：只支持下载到单个文件，只支持下载一个table或一个partition。
* 预扫描数据：上传之前可先预扫描本地文件，通过表schema对数据内容进行校验。
* session管理：每次上传、下载会生成一个对应的session目录，保存本次上传下载的状态信息。
* 续传：只支持上传的续传，因为网络或tunnel服务的原因出错，支持续传。
* 历史信息：可以通过show命令查看session相关信息，用purge命令删除session。
* 脏数据：如果设置discard-bad-records=true，脏数据文件会保存在session目录中

## FT测试case

目的：endtoend把常用的上传下载测试跑通，包含所有的数据类型，数据正确性通过。

review代码：<http://reviewboard.aliyun-inc.com/ReviewBoard/r/91165/>

Schema：create table table\_name(i1 bigint, s1 string, d1 datetime, b1 boolean, doub1 double, de1 decimal) partitioned by(p1 string, p2 string); alter table table\_name add partition(p1="a1",p2="a2");

### 测试一：文件上传下载，test\_file\_upload\_download

* 上传3M文件成功。
* 比较正确性：比较总行数及整形\decimal类型列的校验和。
* 下载：比较上传文件和下载文件的内容一致。

### 测试二：目录上传及正确性验证，test\_folder\_upload\_validate

* 上传目录，目录中包含100个文件，由2-3M文件组成。
* 比较正确性：总行数及整形\decimal类型列的和。

### 测试三：目录上传及下载，test\_folder\_upload\_download

* 目录上传成功:目录由几个小文件组成
* 下载上传的数据，比较下载数据和预期一致。

### 测试四：脏数据文件上传下载，test\_bad\_folder\_upload\_download

* 测试包含脏数据文件的目录上传，几个文件中都包含脏数据
* 导出脏数据，脏数据和预期一致
* 下载上传的数据，下载数据和预期一致。

### 测试五：特殊类型数据上传下载，test\_data\_type\_upload\_download

* 测试数据包含：null值，布型包含true\0\1\false（包括大小写），double型包含无穷大、无穷小。
* 比较正确性：行数及校验和
* 下载数据，符合预期

### 测试六：字符编码GBK及bom utf测试

* gbk文件上传、下载，数据正确性符合预期
* bom文件上传、下载，数据正确性符合预期

## UT测试case

目的：覆盖到文件上传下载的所有场景。

代码Review: <http://reviewboard.aliyun-inc.com/ReviewBoard/r/91009/>，见src/test/java

### Mock测试:

模似文件上传下载的过程，检查上传下载过程中的状态是否符合预期。

#### 文件上传：FileUploadTest

1. testUploadSingleFile

测试上传一个文件成功，上传成功，检查session 状态符合预期

1. testUploadSingleFileWithStep

测试文件上传成功，检查上传过程中每一个block的状态符合预期信息

1. testUploadSingleFileWithScanStep

测试文件上传成功，指定预扫描，检查上传过程中每一个block的状态符合预期

#### 文件上传，出错自动重试: FileUploadRetryTest

1. testUploadFailRetry5

测试上传文件时，同一个block上传出错5次，正常上传

1. testUploadRetry6Fail

测试上传文件时，同一个block上传出错6次，上传失败

#### 目录上传：FolderUploadTest

1. testUploadFolder

测试上传一个目录成功，保存的session 状态符合预期

1. testUploadFolderWithStep

测试目录上传，上传过程中，检查每一个block的状态信息符合预期

1. testUploadFolderWithScanStep

测试目录上传，指定预扫描，检查上传过程中每一个block的上传状态符合预期

1. testUploadNullFolder

测试上传一个空目录，上传空目录会失败

1. testUploadFolder2000Files

创建一个有2000个文件的目录，mock测试上传一个目录成功，且blockid和文件的顺序符合预期

#### 断点续传： UploadResumeTest

1. testResumeFileUpload

测试从文件头位置开始文件续传，上传成功状态符合预期

1. testResumeFilePositionUpload

测试文件续传，从某一行开始,上传成功状态符合预期

1. testResumeFolderUpload

测试目录续传，从某一个文件\某一个block\某一行开始续传, 上传成功状态符合预期

1. testResumeFolderWithFileIndexUpload

测试目录续传, 按续传的文件index顺序开始续传, 上传成功状态符合预期

1. testResumeFolderWithScanTrueFileIndexUpload

测试目录续传:

\* 测试强制续传

\* 当前续传的状态是scan

\* 先预扫描完文件，再resume

\* 测试续传过程中状态符合预期

#### 脏数据文件和目录上传：BadRecordsFileUploaderTest

1. testFailDiscardBadRecordsFalse

测试--discard-bad-records=false，不丢弃脏数据，当有脏数据时出错

1. testDiscardBadRecordsTrue

测试--discard-bad-records=true，当有脏数据时，upload执行成功

1. testDiscardBadRecordsMaxSize

测试最大discard bad records的边界，并检查脏数据信息 如果大于max-bad-records，则出错，如果小于，则导入数据,把数据存到bad.txt文件中

1. testFolderDiscardBadRecordsTrue

测试上传目录成功，脏数据行数及内容符合预期

1. testFailFolderDiscardBadRecordsTrue

上传目录失败，显示脏数据文件，且脏数据文件内容符合预期

1. testOnlyScanFolderDiscardBadRecordsTrue

测试预扫描目录成功，脏数据行数及内容符合预期

#### 数据下载： FileDownloaderTest

1. testDownloadFile

测试下载数据，包括所有的数据类型。

测试目的：

\* mock下载数据的过程，把数据下载到临时文件。

\* 比较下载数据文件的内容，和原文件一致。

1. testDownloadFileWithCrash4\_Success

测试下载，当下载出错时，下载行为符合预期。

测试目的：

\* 模拟出错4次, downloader retry成功。

\* 下载文件内容正确

1. testDownloadFileWithCrash5\_Fail

测试读数据出错5次，下载失败

测试目的：

\* 读数据时，连接出错5次，下载失败，异常信息符合预期.

### 文件解析及数据转换

#### 文件解析： TextFileReaderTest

1. testReadLineOneline

测试文件,读取一行，行分隔符和列分隔符为一个字符

1. testReadLineMoreLine

测试读文件，读取多行, 行分隔符和列分隔符为一个字符

1. testReadLineChinese

测试读文件，单个字符分隔符，文件内容包括中文，编码格式为utf8

1. testReadLine\_more\_char\_split

测试读文件，行分隔符和列分隔符为多个字符

1. testReadLineMoreCharChinese

测试读文件，行分隔符和列分隔符为多个字符，内容包含中文

1. testReadLineChineseSplit

测试行分隔符和列分隔符为中文字符

1. testReadLineChineseBom

测试文件内容包含中文，文件带有bom头

1. testFieldDelimiterA

测试单个字符A行分隔符，字符A和正则表达式无关

1. testSpliter

测试字符串split，包括行分隔符为正则表达式的特殊字符

1. testSkipAndRead

测试skip到文件的位置，续传读取下一行. 包括bom文件和非bom文件

1. testReadLineMoreCharChineseWithGBK

测试读文件，多个字符分隔，内容包含中文，编码格式为gbk

1. testRecordDelimiterIncludeFieldDelimiter

测试行分隔符包含列分隔符，正常执行

#### 数据类型转换：RecordConverterTest

1. testNormal

测试所有正确的数据类型由string数组<->record的相互转换。

数据类型包括：BIGINT\STRING\DATETIME\BOOLEAN\DOUBLE\DECIMAL

测试目的：

\* 数据上传需要把string数组转成tunnel的一个record.

\* 下载需要把rocord转换成string数组。

1. testBoolean

测试布尔型，数据类型的取值："true", "false", "1", "0"

测试目的：

\* boolean型上传时，数据文件能够包括这4个值。

\* 上传数据如果不在这4个值内，抛出脏数据异常，脏数据异常信息符合预期

\* 下载时，对应的值为"true", "false"

1. testDouble

测试double类型取值 取值范围：数字、科学计数法、无穷大（"Infinity"）,无穷小（"-Infinity"）

测试目的：

\* double类型的值可以在此取值范围内

\* 不在此范围内值，抛出脏数据异常，异常信息符合预期

\* 下载值和上传值一致。

1. testString

测试字符类型

测试目的：

\* 字符串长度小于2M字符，正常

\* 大于2M字符串，抛出脏数据异常，异常符合预期

\* 下载字符串一致

1. testDatetime

测试日期类型，及pattern出错的情况

测试目的：

\* 指定pattern正确，数据正常转换

\* 指定pattern和数据不一致，抛出脏数据异常，异常符合预期

\* 上传和下载数据一致

1. testDecimal

测试decimal类型

测试目的：

\* 上传下载数据一致

\* 脏数据出错，出错符合预期

1. testNull

测试Null indicator值

测试目的：

\* 测试所有数据类型的空值上传和下载一致

1. testBigint

测试bigint的最大值，最小值的边界, 最大值：9223372036854775807， 最小值：-9223372036854775807

#### session管理：SessionHistoryManagerTest

1. testListHistory

测试session按session的创建时间排序 .

测试目的：

\* 测试session history是否按设置的create time进行排序

1. testGetlastestSession

测试取默认的session

测试目的：

\* 测试取最后一个生成的session，即最近的session

### 命令行解析

#### 上传命令：ParseUploadCommandTest

1. testProjectTablePartition

upload命令，表名包括project.table/partition格式

命令： upload src/test/resources/test\_data.txt up\_test\_project.test\_table/ds='2113

1. testProjectTable

upload命令，表名包括project.table格式

命令： upload src/test/resources/test\_data.txt test\_table/ds='2113

1. testTable

upload命令，只包括table

命令： upload src/test/resources/test\_data.txt test\_table

1. testTablePartition

upload命令，包括table/partition格式

命令： upload src/test/resources/test\_data.txt test\_table/ds='2113',pt='pttest'

1. testFailTablePartition

命令格式不对的测试table/partition,格式不对

命令： upload src/test/resources/test\_data.txt test\_project.test\_table/ds/xxx

命令： upload src/test/resources/test\_data.txt test\_project.test\_table.partition

1. testFileSourceFileNoExist

命令行源文件不存在

命令： upload src/test/resources/test\_data\_noexist.txt test\_project.test\_table

1. testFailMoreArgs

命令行参数过多

命令： upload src/test/resources/test\_data\_noexist.txt test\_project.test\_table/ds='2113',pt='pttest' badcmd

1. testFailLessArgs

命令行参数过少

命令： upload src/test/resources/test\_data\_noexist.txt

1. testOptionsNormal

测试option参数正常的情况，所有option参数通过命令行传递

1. testOptionsDiscardBadRecords

测试--discard-bad-records的值，只能包括true|false

1. testOptionsScan

测试--Scan的值，只能包括true|false|only

1. testUploadOptions\_charset

测试指定字符编码gbk|utf8|gb2312解析，及不支持的编码

1. testOptionsMix

测试命令行->配置文件->默认值的优先级顺序

1. testOptionsQuote

测试命令行解析, option用"="号分隔

如命令：upload src/test/resources/test\_data.txt test\_table -rd="\r\n"

1. testOptionssplit

测试命令行解析, option用空格分隔

如：upload src/test/resources/test\_data.txt test\_table -rd "\r\n"

1. testShortCommand

测试短命令格式

如命令： upload src/test/resources/test\_data.txt test\_table -c "gb2312"

1. testUploadCommandLine

测试的命令行, 命令行需要保存在context中

1. testSplitWithRNT

测试行分隔符、列分隔符，特殊字符\r\n\t的解析

原因，命令行会把转义字符作为字符串传入 如：upload test\_data.txt test\_table -rd "\\r\\n"

1. testOptionsEndpointIdKey

测试使用endpoint\id\key

命令：upload src/test/resources/test\_data.txt test\_table -c "gb2312" -e http://service.odps.aliyun.com/api -i xxx -k xxx

1. testFailMoreoption

测试不存在的option

命令：upload test\_data.txt test\_table "--badoption=xxx"

1. testFailDelimiterInclude

测试列分隔符，不能够包含行分隔符，出错

命令：upload test\_data.txt test\_table -rd "||" -fd "||||"

1. testFailDataformatpattern

测试date format pattern设置出错

命令: upload test\_data.txt test\_table -dfp "abcd"

#### 下载命令：ParseDownloadCommandTest

1. testProjectTablePartition

download命令，表名包括project.table/partition格式

命令：download src/test/resources/test\_data.txt up\_test\_project.test\_table/ds='2113

1. testProjectTable

download命令，表名包括project.table格式

命令： download test\_table/ds='2113 src/test/resources/test\_data.txt

testTable

download命令，只包括table

命令： download test\_table src/test/resources/test\_data.txt

1. testTablePartition

download命令，包括table/partition格式

命令： download test\_table/ds='2113',pt='pttest' src/test/resources/test\_data.txt

1. testFailTablePartition

命令格式不对的测试, table/partition,格式不对

命令： download test\_project.test\_table/ds/xxx src/test/resources/test\_data.txt

命令： download test\_project.test\_table.partition src/test/resources/test\_data.txt

1. testFailMoreArgs

命令行参数过多

命令： download src/test/resources/test\_data\_noexist.txt test\_project.test\_table/ds='2113',pt='pttest' badcmd

1. testFailLessArgs

命令行参数过少

命令： download src/test/resources/test\_data\_noexist.txt

1. testOptionsNormal

测试option参数正常的情况，所有option参数通过命令行传递

1. testdownloadOptions\_charset

测试指定字符编码gbk|utf8|gb2312解析，及不支持的编码格式异常退出

1. testOptionsMix

测试命令行->配置文件->默认值的优先级顺序

1. testOptionsQuote

测试命令行解析, option用"="号分隔

如命令：download test\_table src/test/resources/test\_data.txt -rd="\r\n"

1. testOptionsSplit

测试命令行解析, option用空格分隔

如：download test\_table src/test/resources/test\_data.txt -rd "\r\n"

1. testShortCommand

测试短命令格式

如命令： download test\_table src/test/resources/test\_data.txt -c "gb2312"

1. testdownloadCommandLine

测试的命令行，命令行需要保存在context中

1. testSplitWithRNT

测试行分隔符、列分隔符，特殊字符\r\n\t的解析

原因，命令行会把转义字符作为字符串传入 如：download test\_table test\_data.txt -rd "\\r\\n"

1. testOptionsEndpointIdKey

测试使用endpoint\id\key

命令：download test\_table src/test/resources/test\_data.txt -c "gb2312" -e http://service.odps.aliyun.com/api -i xxx -k xxx

1. testFailMoreoption

测试不存在的option

如命令：download test\_data.txt test\_table "--badoption=xxx"

1. testFailDelimiterInclude

测试列分隔符，不能够包含行分隔符，出错

如命令：download test\_table test\_data.txt -rd "||" -fd "||||"

1. testFailDataformatpattern

测试date format pattern设置出错

如命令: download test\_data.txt test\_table -dfp "abcd"

1. testFilePathIncludeSpaceChinese

测试上传文件路径包括空格和中文

1. testUpperOption

测试命令行option有大写的情况，包含大写会出错

1. testOptionValueNullString

测试option参数值为空字符串的情况，但NULL\_INDICATOR的值可以为空串

#### 续传命令：ParseResumeCommandTest

1. testResumeCommand

正常解析resume命令

1. testResumeCommandForce

测试resume命令，指定force选项

1. testResumeCommandFail

测试resume命令解析出错，出错符合预期

#### 帮助命令：ParseHelpCommandTest

1. testHelpCommand

测试help命令的解析，判断解析后的命令项

1. testHelpCommandFail

测试不存在的命令的帮助，出错信息符合预期

1. testShowHelp

测试执行help命令正常

#### config命令及配置文件：ParseConfigTest

1. testConfigfileNormal

指定配置文件，配置文件配置正常

测试目的：

\* 通过命令行参数，指定配置文件，配置文件内容符合预期

1. testPathWithChineseSpace

指定配置文件，路径有中文件和空格

测试目的：

\* 通过命令行参数指定配置文件，配置文件路径包括中文字符及空格，配置项符合预期。

1. testOptionValueWithSpace

指定配置文件，配置文件中值字符串的前后有空格

测试目的：

\* 指定配置文件，配置项前后有空格，需要trim掉

\* FIELD\_DELIMITER和RECORD\_DELIMITER不能够trim，因为可能包含\r\n等

1. testOdpsConfigfile

指定配置文件，使用新console的配置 id命名为access.id

测试目的：

\* 配置文件使用新console的配置，配置项符合预期

1. testDefaultvalue

测试工具的默认值

测试目的：

\* 测试未指定命令行，配置文件内容也为空，工具的默认值符合预期

\* 测试指定了命令，优先级符合预期

1. testSettings

测试upload的setting项

测试目的：

\* 测试配置文件中配置的setttin项的值，有max-bad-records\block-size

1. testConfigCommand

测试config命令解析成功

测试目的：

\* 测试config命令指定配置项，命令解析成功

1. testConfigCommandWithRTN

测试特殊字符\r\n\t解析成功

测试目的：

\* 测试命令行设置转意字符\r\n\t，能够正确配置。

1. testConfigCommandFail

测试config命令解析出错，正确退出

测试目的：

\* 几种情况，config命令出错，正确退出。

#### session删除命令：ParsePurgeCommandTest

1. testPurgeCommand

测试purge命令项的解析

1. testPurgeFail

测试purge命令项解析出错，出错信息符合预期

#### show命令：ParseShowCommandTest

1. testShowCommand

测试show命令正常执行

1. testShowCommandShort

测试show命令短命令格式的执行

1. testShowHistoryCommandFail

测试执行show history命令失败，异常信息符合预期

1. testShowLogBadCommandFail

测试执行show log\show bad命令失败，异常信息符合预期

#### 命令行和partition\_spec的解析：ParsePartitionSpecTest

1. testPartitionSpec

测试正确设置project.table/partition格式

1. testPartitionSpecValueWithSpace

测试partition中包含空格，正常

1. testPartitionCommaWithSpace

测试partition之前逗号前有空格

1. testFailIllegalPartition

测试partition格式非法

## 4T数据及环境测试

测试目的：验证工具的性能及运行环境，及传大数据数据正确性。

准备：

* 数据表结构：create table big(i1 int, i2 int, d1 datetime,b1 boolean, s1 string);
* 数据特征：第一列从1开始自增，第二列为33开始自增，其它几列为固定值
* 数据正确性验证：验证总行数，及第一列和第二列的和（正确性验证通过）
* 传输速度：按大B表示
* 测试环境：对外集群和dailyrun环境

### 4T数据测试

#### 测试一：单独4T数据上传(失败)

* 共4T数据：一个目录，包含2000个软链接文件，指向单个2GB文件。
* dailyrun环境
* 开始时间：2014-04-04 23:06:06
* 结束时间：2014-04-05 23:08:26(upload超过24小时，失败)
* 上传失败：失败原因，一个upload有效期只有24小时

#### 测试二：并发4T数据上传（成功）

* 共4T数据：单个文件2GB, 一个目录，250个软链接文件。
* 耗时：一个upload耗时7小时35分
* 采用压缩，8个并发上传
* 平均速度: 约19MB/s
* 测试环境：dailyrun

#### 测试三：单独500G上传

* 数据大小，单独上传800G
* 耗时: 共9小时
* 传输速度：约26M/s
* 未压缩
* 测试环境：dailyrun

#### 测试四：多个并发，打满网卡

* 10个并发，每个并发传500G数据
* 未压缩
* 测试环境千兆网卡
* 结果：每个进程平均11MB/s，总传输速度110M/s，符合预期

### 外网环境测试

#### 测试一:从外网传500M数据

* 数据大小： 500M
* 对外集群：http://dt.aliyun.com/api
* 耗时：550秒
* 平均：0.92MB/s
* 家庭网络：10Mb带宽
* 下载：文件大小一致，文件内容一致

#### 测试二：从外网走压缩传500M数据

* 走vpn压缩上传
* daily测试环境
* 500M数据，耗时共15秒，平均33MB/s
* 50G数据，耗时27分钟，平均31MB/s（一个目录，100个软链接文件，指向500M数据）

注：压缩效率比较高，中文字符比较多，单条记录内容重复度不高，但多条记录的重复度比较高。

#### 测试三：从外网走压缩，纪录重复度比较低

* 500M数据，压缩上传
* 数据特征: 上一个测试最后一个string改成变量
* 耗时: 250秒
* 平均速度：2MB/s(貌似比较合理)

### windows\linux环境测试

#### 测试一：配置文件，路径中文字符及空格

windows下和linux下都支持

#### 测试二：命令行行列分隔符包含\r、\n或\t转意字符

windows下和linux命令行支持\r\n\t输入，其它转意字符不支持。