### hdfs

HDFSを探索したり、クエリ結果をファイルとして転送することができます。

#### 構文

ファイル一覧の表示

hdfs PROFILE {ls|lsr} PATH

ファイル内容の読み取り

hdfs PROFILE cat [format=csv|json|sequence] [limit=INT] [offset=INT] PATH

特定レコードのフィールド値をテキスト、CSV、JSON形式でファイル転送

hdfs PROFILE put [format=csv|json] [limit=INT] [offset=INT] [partition=t] PATH

特定レコードのフィールド値をHDFSシーケンス形式でファイル転送

hdfs PROFILE put format=sequence [compression\_type=block|record] [fields=FIELD,...] [key\_type=HDFS\_TYPE] [key\_field=KEY\_FIELD] [value\_type=HDFS\_TYPE] [value\_field=VALUE\_FIELD] PATH

必須パラメータ

**PROFILE**

HDFS接続に使用する接続プロファイル

プロファイルはWebコンソールで構成できます。ENT-3.10.2009.0、SNR-3.1.2008.0配布バージョン以降、HDFSプロファイルは接続プロファイルに統合されました。

**{ls|lsr}**

lsコマンドはPATHで指定したパスのファイル一覧を表示します。lsrはPATHで指定したディレクトリおよびその配下すべてのディレクトリを再帰的に探索します。

**cat**

HDFSファイルシステム上のテキストファイル、CSVファイル、JSONファイル、HDFSシーケンスファイルの内容を参照します。formatオプションで指定したファイル形式に従ってパースします。

* テキストベースファイルの場合、内容をlineフィールドに出力します。
* CSV形式の場合、最初の行をフィールド行として認識します。データ内に改行が含まれていても、CSVルールに従えば複数行を1つのフィールド値として認識します。
* JSON形式の場合、ファイルを行単位で読み込みパースします。
* HDFSシーケンス形式の場合、HDFSのWritable実装をログプレッソ型（Javaのデータ型）に変換し、レコード単位で読み込みます。変換形式はログプレッソのHDFSデータ変換型を参照してください。
* キーフィールド名はkeyに指定されます。キーは元の型に関係なく文字列に変換されます。
* 値フィールド名は値の型によって異なるフィールドが使用されます。
* MapWritable型の場合：内部のキー・値マッピングが返される行のフィールドとして返します。Hadoopに内蔵されたWritable実装をログプレッソ型に変換して出力します。
* MapWritable型でない場合：valueフィールドに出力します。

**put**

fieldsオプションで指定したフィールド値をHDFSファイルシステムにファイルとして転送します。ファイルはPATHで指定したパスに作成されます。

* テキスト形式の場合、fieldsオプションで指定したフィールド値を転送します。フィールドを指定しない場合はlineフィールドの値を転送します。
* CSV形式の場合、1行目にfieldsオプションで指定したフィールド名リストを記録します。フィールド値がない場合は空文字列で表示します。fieldsオプションを指定しない場合はlineフィールドの値を転送します。
* JSON形式の場合、fieldsオプションで指定したフィールド値のみ転送します。fieldsオプションを省略した場合はすべてのフィールド値を転送します。
* HDFSシーケンス形式の場合、以下の条件を除きシーケンス形式で転送します。変換形式はログプレッソのHDFSデータ変換型を参照してください。
* 転送するレコードのキーまたは値のいずれかが空の場合、その行は転送しません。
* 値と型が一致しない場合、
* string型は文字列に変換します。
* int、long、float、doubleなどの数値型は0に変換します。
* bool型はfalseに変換します。
* 精度を損なわずに変換できる場合は変換後に出力します（例：long型が指定されているがint値が入力された場合、longに変換して出力）。

**PATH**

ファイルの絶対パス

オプションパラメータ

**compression\_type=block|record**

圧縮形式。このオプションがない場合は圧縮しません。recordはレコード単位圧縮、blockはブロック単位圧縮を意味します。

**format=csv|json|sequence**

参照または転送するファイル形式。テキストファイルを参照または転送する場合はこのオプションを省略します。

**key\_type=HDFS\_TYPE**

ログプレッソのHDFSデータ変換型でHDFS型に定義された型を指定します。

**key\_field=KEY\_FIELD**

キーフィールド名。設定しない場合は1から始まるLongWritableカウンターを使用します。

**limit=INT**

ファイルを読み込む際に出力する行数（デフォルト：無制限）。

**offset=INT**

ファイルを読み込む際にスキップする行数（デフォルト：0）。

**partition=t**

tに設定するとマクロを利用してディレクトリパスを指定できます。マクロを利用するとディレクトリやファイル名を時刻に応じて変更できます。

**fields=FIELD,...**

putコマンドでHDFSに転送するフィールドリスト。区切り文字はカンマ（,）です。

**value\_type=HDFS\_TYPE**

ログプレッソのHDFSデータ変換型でHDFS型に定義された型を指定します。

**value\_field=VALUE\_FIELD**

値フィールド名。設定しない場合は全フィールドを1つのMapWritableとして転送します。

#### 説明

catでファイルを読み取る際、PATHにワイルドカード（\*）を使用すると、特定の文字列パターンを含むすべてのファイルを一度に参照できます。

putでファイルを転送する際、partitionオプションをtに設定すると、マクロを利用して時刻に応じてディレクトリやファイルパスを変更できます。パーティションオプションを指定し、パスにマクロを使用しない場合はクエリが失敗します。

利用可能なマクロは{logtime:FMT}と{now:FMT}です。

* {logtime:FMT}：ログ発生時刻を基準にディレクトリやファイル名を付与
* {now:FMT}：現在時刻を基準にディレクトリやファイル名を付与

マクロは中括弧（{ }）で囲んで入力します。入力例は使用例7番を参照してください。

ログプレッソはJava標準データ型およびIPアドレスなどログプレッソで定義されたデータ型を使用します。HDFSからデータを取得または転送する際、HDFSデータ型に合わせて変換処理を行います。型ごとのデータ変換は以下の表を参照してください。

**ログプレッソとHDFSデータ変換型**

|  |  |  |
| --- | --- | --- |
| ログプレッソ型 | HDFS型 | 説明 |
| string | Text | 文字列 |
| null | NullWritable | null |
| bool | BooleanWritable | ブール（真/偽） |
| int | IntWritable, VIntWritable | 4バイト（32ビット）整数 |
| long | LongWritable, VLongWritable | 8バイト（64ビット）整数 |
| float | FloatWritable | 単精度浮動小数点（single） |
| double | DoubleWritable | 倍精度浮動小数点（double） |

#### 使用例

vmというプロファイルで接続し、ルートパスのファイル一覧を表示

hdfs vm ls /

出力されるフィールドは以下の通りです。

* type（文字列）：ディレクトリの場合はdir、ファイルの場合はfile
* name（文字列）：ファイル名
* path（文字列）：ファイルの絶対パス
* replication（整数）：レプリカ数、ディレクトリの場合は0
* file\_size（整数）：ファイルサイズ、ディレクトリの場合は0
* block\_size（整数）：ブロックサイズ、ディレクトリの場合は0
* modified\_at（日付）：最終更新日時
* permission（文字列）：権限設定
* owner（文字列）：所有者
* group（文字列）：所有グループ

vmプロファイルで接続し、/tmp/LICENSE.txtファイルの最初の1行をスキップして5行を参照

hdfs vm cat offset=1 limit=5 /tmp/LICENSE.txt

vmプロファイルで接続し、/tmp/malware.csvファイルから3行を参照

hdfs vm cat format=csv limit=3 /tmp/malware.csv

vmプロファイルで接続し、/tmp/iis.jsonファイルから1行を参照

hdfs vm cat format=json limit=1 /tmp/iis.json

vmプロファイルで接続し、/tmp/classloading.seqファイルから2レコードを参照

hdfs vm cat format=sequence limit=2 /tmp/classloading.seq

JMXクラスローディングログのうち、UnloadedClassCountとLoadedClassCountのみを/tmp/class.txtパスに出力

table classloading | hdfs vm put fields=UnloadedClassCount,LoadedClassCount /tmp/class.txt

sys\_cpu\_logsログを/tmp配下の日付別ディレクトリに出力

table sys\_cpu\_logs | eval line=concat("idle: ", idle, ", kernel: ", kernel, ", user: ", user) | hdfs vm put partition=t /tmp/{logtime:yyyyMMdd}/cpu.txt

JMXクラスローディングログのうち、LoadedClassCount、UnloadedClassCount、TotalLoadedClassCountを出力

table classloading | hdfs vm put format=csv fields=LoadedClassCount,UnloadedClassCount,TotalLoadedClassCount /tmp/classloading.csv

JMXクラスローディングログをJSONファイルとして出力

table classloading | hdfs vm put format=json /tmp/classloading.json

JMXクラスローディングログ全体をHDFSシーケンスファイルとして出力

table classloading | hdfs vm put format=sequence /tmp/classloading.seq

JMXクラスローディングログのうち、LoadedClassCount値のみを出力

table classloading | hdfs vm put format=sequence value\_type=long value\_field=LoadedClassCount /tmp/classloading.seq