DriveStat Administration Guide

J. Fromm, K. Genser

December 28, 2001
version v1_3

Contents

Chapter 1

Overview

1.1 Introduction

`drivestat` is a package that collects tape drive usage statistics in a centralized location.

In the Fermilab installation the data is being collected into an ORACLE database which resides on a MISCOMP[1] computer and can be accessed using standard ORACLE methods including but not limited to the MISWEB[2] forms pointed to by the link on the `drivestat` WWW page:
`http://www-isd.fnal.gov/drivestat/`

Two servers are used to acomplish the task of collecting the information. One server (ds_server) listens for user requests and transforms them into "request files" whereas the other server (dbinter) process the request files and interacts with an (Oracle) database.

This document describes the administration of the servers.

1.2 Prerequisites

It is assumed that the servers run on a computer where an ORACLE database resides. Fermilab UPS/UPD[3] framework is used to distribute, install, start and stop `drivestat` servers. The servers are written in Python and rely on the python_dcoracle interface with both of them avaliable through the ups/upd mechanizm.

Since the servers are written in Python and use embeded SQL they should in principle be portable to a different environment than the one for which they were created.

Chapter 2

Basic Drivestat Administration

2.1 Configuration

`drivestat` uses one configuration file located in `$DRIVESTAT_ROOT/config/drivestat.cfg`. Each line of the file is of the form:

```
keyword=value
```

.

The following is a list of keywords that need to be in the file:

- port - The port number which the server listens for connections.
- host - The hostname that the server is running on.
- ora_user - The Oracle user name.
- ora_pw - The Oracle password. (This is not a Unix password).
- ora_instance - The Oracle instance (prd1 for the production version).
- sleep_interval - This is the number of seconds that the dbinter process sleeps before waking up to check for new requests.
- debug_level - The variable controlling the level of log output generated by the servers. It takes intiger values with 0 being a ballanced compromise between a terse and verbose level.
- logger_path - The location of the server log files together with the name of the log file. The name will be appended with _YYYYMMDD suffix to make sure the size of the log files is manageable.
- request_directory -

  The location where the `drivestat` server will cache it's requests and where processed requests are stored. The ds_server simply puts the requests into this location, allowing the database interface (dbinter) process to scan this file to perform the actual requests.

  The following subdirectories need to be created in that subdirectory:

- todo - this is where the request files are placed for processing
- done - place where processed request files are stored
- bad - place where request files causing erros are stored
- totar - directory used while processing done files
- tar - directory where done files are stored after being tar'ed on a daily basis

An example configuration file is supplied below and included in $DRIVESTAT_DIR/templates/config directory

```
port=5001
host=node.fnal.gov
ora_user=oracle_user
ora_pw=oracle_password
ora_instance=dev1
logger_path=/drivestat/log/serverlog.out
request_directory=/drivestat/requests
sleep_interval=60
debug_level=0
```

The directory structure within the $DRIVESTAT_ROOT directory can be created by making a copy of files located in the product templates directory:

```
cp -r $DRIVESTAT_DIR/templates/* $DRIVESTAT_ROOT
```

The directory strucure in the templates directory assumes that the configuration file, log files and request files are all stored in the same directory tree. The configuration file has to be modified accordingly to reflect the specifics of the particular instalation.

It is worth mentioning that the port number is used for tcp connections between the clients reporting on the drives and the ds_server. The change of the port number requires corrsponding changes for all the clients. (It is defined in the ups product table file).

2.2 Server Startup/Shutdown

There are two Python processes that need to be started on the machine where the servers are to be run:

- $DRIVESTAT_DIR/bin/ds_server.py
- $DRIVESTAT_DIR/bin/dbinter.py

In order to run them, one should issue the command:

```
> ups start drivestat
```

To gracefuly shut down the `drivestat` servers one should issue the command:

```
> ups stop drivestat
```

It is expected that those commands should be included in the ups startup/shutdown configuration for the node on which the servers run.

2.3 Database Maintenance

There are four database tables used by `drivestat`:

- MASTER_DRIVESTAT
- DRIVESTAT_LOG
- DRIVE_MAINTENANCE_LOG
- DRIVE_LOCATION

These tables can be created using the script:
`$DRIVESTAT_DIR/src/db/create_tables.sql`. This script first drops the tables if they exist, and then creates them. The content of the tables is described in the `drivestat` users guide.

Chapter 3

Installation

3.1 Server Installation

Installation of the `drivestat` server involves installing the ups `drivestat` product, tailoring the product, setting up the configuration file, crating the log and request directories (described in the previous chapter), creating the database tables, and starting the servers.

The following is an example of installing and declaring the `drivestat` product as a test instance of the server on a node running a Sun version of Unix:

```
> upd install -z /fnal/ups/db -f SunOS+5 -q server drivestat v1_3
> ups declare -t -z /fnal/ups/db -f SunOS+5 drivestat v1_3
```

Subsequently the product needs to be tailored:

```
> ups tailor -t -O $DRIVESTAT_ROOT drivestat
```

and the previously defined directory structure needs to be created within the $DRIVESTAT_ROOT directory.

3.1.1 Creating the Database Tables

On the server machine, issue the following commands:

```
> setup prd0
> cd $DRIVESTAT_DIR/src/db
> sqlplus ora_user/ora_pass @ create_tables
```

This example assumes that oracle username is “ora_user” and the password is “ora_pass”. It also assumes that the database instance is “prd0”.

Bibliography

[1] Fermilab Computing Division Library Document WP0067: MISCOMP Project

[2] Fermilab Computing Division Library Document PU0397: MISWEB

[3] Fermilab Computing Division Library Document PU0117: UPS UNIX Product Support