SAM installation for dummies

Tomasz Wlodek

University of the Great State of Texas at Arlington

Abstract: I give "how-to" instructions describing installation of a SAM station at a remote site. Those instructions do not cover the installation of SAM station at FNAL.

Where to get more information:
About SAM installation: http://d0db.fnal.gov/sam/doc/userdocs/station_setup_procedure.html
About UPS/UPD: http://www.fnal.gov/docs/products/ups/

Before you start: Here is a list of things you should know/do before you start installing SAM station:

1. UPS/UPD should be installed on your machine. (The instruction can be found on the DØRACE home page; http://www-hep.uta.edu/~d0race. Setup instruction for Phase I covers this issue).
2. Create an account "sam" with uid 7816. If you do not know how to create account ask your system administrator.
3. Create a subdirectory ~sam/private pointing to a large disk area for storage of SAM internal files.
4. Your machine should have an account from which you install new products. Usually this is either root or "products" account (with needed permissions). You will have to have access to this account.
5. You should be familiar with UPS/UPD and understand how to use it.
6. Make sure that the UPS version is v4_6 (or later) as of Feb. 5, 2002. If not, you should upgrade it. I do not know why, but it does not work properly with earlier versions.
7. Write to sam-admin@fnal.gov and ask them to register your station. Give them the names of the nodes on which your SAM station will run. Give them your station name. The station name isof your choose: for example uta-hep, uta-analysis or something like that.
8. You will be given a "magic string" – a secret string of characters - which your station will use for identification purposes. Save it in temporary file or write down on a paper, you will need it later.

Ok, now let us start. Firs of all, login to the products maintenance account and install SAM client software:

```
setup upd
upd install sam -q prd -G "-c"
upd install sam -G "-c"
```

I assume that you do not need development version and you are content with "current" release. Now install SAM server.

```
upd install sam_bootstrap -G "-c"
```

Now you have to configure the server ("tailor") to suit your needs.

The following steps must be executed for each node of your SAM station which will run sam.

```
ups tailor sam_bootstrap
```

The program will ask you several questions.

First it will give you a few questions about where the SAM home area is located (I suggest ~sam/private), who should receive mail messages from SAM in case of troubles (I suggest: send it to user sam and from there set up an automatic forwarding to whom it is necessary). If you do not know what to reply to a particular question agree to the default.

Then it will be asking you questions whether or not it should install particular servers. There are several modules which make SAM. Experts can install all of them, you normally need only four: station, stager, fss and bbftp. So when computer asks you if you would like to install any of these four say yes and no to all others.

For each module you choose to install you will be asked whether you want production or development version: say production. You will be asked which sam_station version you should install, usually the program knows by default which is the right one. Then it will ask you for the station name. (You have chosen the name in preliminaries). An example log of this step you will find in http://d0db.fnal.gov/sam/sam_bootstrap.html, have a look at it.

At the end you will be asked to execute command

```
ups tailor sam_bbftp <bb_ftp version>
```

from products account and after that execute from sam account:

```
ups installAsSam sam_bbftp <bbftp version>
```

```
ups installAsSam sam_bbftp
```

This command will ask you for the "magic string" and write it to the secure area.
After that you will be asked to execute a particular "cp" command to move some files to the sam area. Execute it.

The steps starting from "ups tailor sam_bootstrap" until this point should be executed at every node of your station you want to run sam.

Starting and stopping sam station

From the sam account execute
```
ups start sam_bootstrap
```
or
```
ups stop sam_bootstrap
```

This must be done separately at every node you run sam station.

Configuring SAM station:

Before you can import files from Sam you have to define disk area to be used by SAM and define the groups. This is done by executing the following commands from account sam:

```
setup sam
sam add disk --station=uta-analysis --mount=/scratch/sam/ --
sizeK=14000000
```

It is understood that you should give your station name and the area on your station where you want to mount the files. Afterwards you have to declare a group:

```
sam add group --group=np --max_disk=2g --max_projects=4 –
admin=tomw,sam –fair_share=1
```

You should give the group of your choice and the administrator Id on your station.

How to import data files from SAM to your station

First you have to define a dataset

```
sam define dataset --defname="test42" --group=np --dim='file_name
your filename'
```

This defines a dataset called "test42" (the name is of your choice) which consists of the file "*your filename*". Once this is done run:

```
sam run project get_file.py –interactive
```

This will run interactively the script get_file.py. This script should look:

```python
#!/usr/bin/env python
#
# This file sets up and runs a SAM project.
#
import os, sys, string, time, signal
from re import *
from globals import *
import run_project
from commands import *
##################################################################
#
#   Set the following variables to appropriate values

# Consult database for valid choices
sam_station       = "uta-analysis"

# Consult Database for valid choices
project_definition = "test42"

# A particular snapshot version, last or new
snapshot_version   = 'new'
```

```
# Consult database for valid choices
appname          = "test"
version          = "1"
group            = "np"


# The maximum number of files to get from sam
max_file_amt     = 1

# for additional debug info use "--verbose"
##verbosity        = "--verbose"
verbosity         = ""

# Give up on all exceptions
give_up          = 1


def file_ready(filename):
        # Replace this python subroutine with whatever you want
to do
        # to process the file that was retrieved.
        # This function will only be called in the event of
        # a successful delivery.
        print "File ",filename," has been delivered!"
#
        os.system('cp '+filename+' … copy if whenever you would
like it to go )
        return
```

Appendix: Useful UPS/UPD commands:

`upd list -acK+ <product>` : will give you the available versions of a given product.