

Chapter 13: Providing Access to AFS Products

This chapter describes how to provide access on your local machine to **UPS** products installed in AFS space.

13.1 Overview

Much of the information in this chapter is adapted from document number TN0091, *Configuring a Local UPS Database (While Still Using the Centrally Supported AFS database)*, found on the Web at <http://www.fnal.gov/docs/TN/tn0091.html>.

To minimize duplicate effort in supporting software, a centrally-supported **UPS** database in AFS space is maintained by the product developers. Systems running AFS are encouraged to use the AFS **UPS** database for the majority of their software needs. However, there are cases where a local database is needed in addition to the AFS database (for locally maintained or developed software, different version requirements, and so on).



A system in AFS space does not need to run the bootstrap procedure documented in Chapter 14: *Installing UPS and UPD from Bootstrap*. **UPS**, **UPD**, and **perl** (these products together are referred to as **CoreFUE**) are already available to you via the **UPS** database in AFS space.

You can configure your system for a number of different options regarding AFS product availability:

- a local **UPS** database but no local **CoreFUE** installation, providing access to local and AFS products
- a local **UPS** database and local **CoreFUE** installation, also providing access to local and AFS products
- no local **UPS** database (nor local **CoreFUE** installation), providing access to AFS products only

Whether you want to maintain a local database or not, if you want access to the **UPS** products in AFS space, you need to update your `/usr/local/bin` area as shown in section 13.5 *Updating /usr/local/bin to Access AFS Products*.

For those of you who choose to maintain a local database, we recommend that you *not* install **CoreFUE** locally unless it is absolutely necessary. In most cases, the disadvantages (extra product maintenance responsibilities and a more complicated configuration) considerably outweigh the benefits (access to products when AFS is down and more flexibility in file naming conventions).



Note that the concepts discussed here are equally applicable to local **UPS** databases on machines in an NIS cluster with its own common NFS-mounted database. You must make suitable modification to the particular details, e.g., wherever you see `/afs/fnal.gov/ups`, replace it with the appropriate path to the NFS-mounted area, e.g., `/fnal/ups`.

13.2 Configuring a Local Database to Work With AFS

Please see document number TN0091, *Configuring a Local UPS Database (While Still Using the Centrally Supported AFS database)*, found on the Web at <http://www.fnal.gov/docs/TN/tn0091.html>.

Here we describe how to configure your system to provide access both to locally installed products, declared in a local **UPS** database, and to products in the AFS-space **UPS** database.¹ In this and following subsections, `$PARENT_DIR` refers to the local directory under which all the **UPS** database files and product files reside.



Note ahead of time that there are several local configurations preset in AFS space (i.e., the AFS `$SETUP_DIR/upsdb_list` file recognizes these locations). We recommend that you choose one of them. In fact, if the local database and products area are put in a non-preset location, then this scheme becomes much harder to implement without a local copy of **UPS/UPD**; see section 13.4 *Additional Steps for Unfamiliar Naming Conventions*. In general, the recommended directory under which all the **UPS** database files and product files reside (`$PARENT_DIR`) is:

<code>/fnal/ups</code>	the standard naming convention provided by several bootstrap configurations for product server nodes
------------------------	--

Other preset configurations:

<code>/local/ups</code>	standard provided by the Fermi RedHat Linux bootstrap for satellite nodes
-------------------------	---

<code>/usr/products</code>	another popular naming convention
----------------------------	-----------------------------------

1. The astute reader will notice that there are an infinite number of alternatives; the steps shown are, however, sufficient for most purposes.

13.2.1 Steps to Create and Configure the Database

1) Create the top-level directory (`$PARENT_DIR`). Make sure that your *products* account (or whichever account should own the product files) can read and write to this directory.

2) Log in as *products*, `cd` to `$PARENT_DIR` and create the following directories:

```
% mkdir db                contains the database
% mkdir db/.upsfiles      local UPS configuration file goes here
% mkdir db/.updfiles     local UPD configuration file goes here
% mkdir prd              local product file hierarchy begins here
% mkdir man              local man pages go here
% mkdir catman           local catman pages go here
```

3) Create the **UPS** database configuration file for your local database:

```
% setup ups              using the copy of UPS in AFS space!
% cd $PARENT_DIR/db/.upsfiles
                           change to the location of your local UPS
                           configuration
% cp $UPS_DIR/ups/dbconfig.template ./dbconfig
                           copy the template dbconfig file from
                           the UPS in AFS space to your area
```

Edit your local `dbconfig` file and replace `/fnal/ups` with your `$PARENT_DIR`, if different from `/fnal/ups`.

4) Create the **UPD** configuration file for your local database:

```
% setup upd              using the copy of UPD in AFS space!
% cd $PARENT_DIR/db/.updfiles
                           change to the location of your local UPD
                           configuration
% cp $UPD_DIR/ups/updconfig.template
                           ./updconfig
                           copy the template updconfig file
                           from the UPD in AFS space to your area
```

In most cases the default `updconfig` file should be perfectly adequate.

- 5) Create the FUE initialization files for your system. These are the files that will be called when users log in (or when other processes start) in order to initialize the FUE environment.

If your configuration follows a well-known naming convention (/fnal/ups, /local/ups, /usr/products or /usr/products/CMSUN1) you can take advantage of the configuration already maintained in AFS space by creating symbolic links that you never need to modify again (here we assume that \$PARENT_DIR is set to /fnal/ups):

```
% cd $PARENT_DIR      change to your $PARENT_DIR
```

```
% ln -s /afs/fnal.gov/ups/etc ./etc
```

this makes your \$SETUPS_DIR a link to AFS

If you are creating the “courtesy links”, you should log in as *root* and run the following commands:

```
% cd /usr/local/etc
```

```
% ln -s /afs/fnal.gov/ups/etc/setupsh ./setupsh
```

```
% ln -s /afs/fnal.gov/ups/etc/setupcsh ./setupcsh
```

Your local database is now configured.



If your configuration does not conform to a well-known convention, please refer to section 13.4 *Additional Steps for Unfamiliar Naming Conventions*.

13.2.2 Post-Configuration: Reinitialize FUE Environment

To use your configured database, reinitialize the FUE environment for your process by running:

For the C shell family: % **unsetenv PRODUCTS**

```
% source $PARENT_DIR/setupsh
```

For the Bourne shell family: \$ **unset PRODUCTS**

```
$ . $PARENT_DIR/setupsh
```

Your \$PRODUCTS environment variable should now include both databases (with the local database coming before the AFS database).

13.2.3 A Note about Product Installation for this Configu-

ration

Users of this type of configuration typically install products using **UPD** (see Chapter 6: *Installing Products Using UPD*). Make sure that the product installers on your system know that to install a product into the local database, they must use the AFS installation of **UPS/UPD**, but **UPD** must use your local configuration files. This is very important! Assuming that `$PRODUCTS` lists your database first, and the product in question doesn't exist in the AFS database, you can run `upd install` without any database option and your product will go into the local database.¹ Otherwise, include the `-z` option in the `upd install` command, e.g.,:

```
% upd install -z /fnal/ups/db[:other-dbs] ...
```

13.3 Installing a Local Copy of CoreFUE



Recall that we discourage installing and maintaining **CoreFUE** locally when the machine is running AFS. The pros and cons are spelled out in document number TN0091, *Configuring a Local UPS Database (While Still Using the Centrally Supported AFS database)*, found on the Web at <http://www.fnal.gov/docs/TN/tn0091.html>.

To install **CoreFUE** locally, first create and configure your local **UPS** database as outlined in section 13.2 *Configuring a Local Database to Work With AFS*. Use the AFS installation of **UPD** to install **UPS**, **UPD** and **perl** into the local database (yes, **UPD** can install itself elsewhere). Then, in your `$SETUPS_DIR/upsdb_list` file (`$SETUPS_DIR` is set in the `dbconfig` file), make sure that you include/activate the line:

```
/afs/fnal.gov/ups/db
```

A Note about Product Installation for this Configuration

Whenever you use **UPD**, set up the instance in the local database to ensure that it uses your local `updconfig` file by default. If you set up the AFS installation of **UPD**, you can use `upd install -z /path/to/yourdb[:other-dbs]` to make it use the local configuration.

1. This also assumes the local `updconfig` file says to install in the local database.

13.4 Additional Steps for Unfamiliar Naming Conventions

If your **UPS** database configuration does not conform to one of the well-known conventions in AFS space, you will need a way of making sure that your local **UPS** database is included in `$PRODUCTS`. There are three ways to accomplish this:

- 1) Lobby to be added to the list of well-known conventions. Send mail to `ups@fnal.gov` stating the name of the local **UPS** database, and a good reason why it should be mentioned in the lab-wide AFS `upsdb_list` (list of known local databases).
- 2) Use `$UPS_EXTRA_DIR`. Make sure that everybody who needs access to your local **UPS** database modifies all of their login scripts (and other scripts) to set the `$UPS_EXTRA_DIR` environmental variable to your database before they source the `setups.[c]sh` script. This is a viable alternative if there is only a small community of people who need this database (e.g., a small group of developers on your local system).

For example:

```
# Cshell example of $UPS_EXTRA_DIR
#
setenv UPS_EXTRA_DIR /our/unfamiliar/local/db
source /afs/fnal.gov/ups/etc/setups.csh
```

Any directories in `$UPS_EXTRA_DIR` will be prepended to the database directories listed in the `upsdb_list` file the first time you source the `setups.[c]sh` script. (Of course, you can always prepend the appropriate database to your `$PRODUCTS` manually at any time).

- 3) Install the components of **coreFUE** locally (**UPS**, **UPD** and **perl**). Maintain your own version of the `setups.[c]sh` scripts by installing a local copy of the **coreFUE** product into your database. Setup the AFS space **UPD** product:

```
% setup upd
```

Use this to install **coreFUE** into your local database and chain it to current:

```
% upd install coreFUE -z $PARENT_DIR/db -G -c
```

Then make sure that your copy of the `$SETUPS_DIR/upsdb_list` contains all of the directories that you wish to include in `$PRODUCTS` (including the AFS **UPS** database).

If you are creating the “courtesy links”, you should log in as *root* and issue the commands (`$PARENT_DIR` is the common parent directory for the local database and products area):

```
% cd /usr/local/etc
% ln -s $PARENT_DIR/etc/setups.csh ./setups.csh
% ln -s $PARENT_DIR/etc/setups.sh ./setups.sh
```

Remember, you will need to keep these local copies of **UPS**, **UPD**, **perl** up to date!

13.5 Updating /usr/local/bin to Access AFS Products

Whether you configure a local **UPS** database or not, if your machine runs AFS and you want access to any AFS-space **UPS** products, you need to update certain files (or links to files) in your local `/usr/local/bin`. These required links and/or files are associated with programming shell and login shell products in the AFS-space **UPS** database. Currently the list of products that require files or links to files in `/usr/local/bin` are: **perl**, **tcsh**, **bash**, **python** (if you need it), and **systools**¹ (see section 9.1 *Installing Products that Require Special Privileges* for more information). We are trying to minimize the number of products which write into `/usr/local`, and we hope the process of updating this area won't be necessary in the future.

Here's how to update your local `/usr/local/bin`:

- 1) Mount `/afs` by running `mount /afs`.
- 2) Set the variable `$PRODUCTS` to your local database.
- 3) You need your local node to point to the **UPS** in AFS space. If you've configured a local database, you've probably already done this step. If not, login as `root` and issue the following commands to set your `/usr/local/etc` courtesy links to point to `/afs/fnal/ups/etc/setups.[c]sh`:

```
% cd /usr/local/etc
% ln -s /afs/fnal.gov/ups/etc/setups.csh .
% ln -s /afs/fnal.gov/ups/etc/setups.sh .
```

- 4) Then, still logged on as `root`, update your `/usr/local`, by running the following commands (from any directory):

```
% source /usr/local/etc/setups.csh
```

(or `$. /usr/local/etc/setups.sh` for Bourne shell)

1. As of this writing, **systools** is not really part of this list, but we expect it to be added.

```
% ups installasroot perl
% ups installasroot bash
% ups installasroot tcsh
% ups installasroot python
```