Fermi National Accelerator Laboratory

FERMILAB-Conf-91/174

UPS UNIX Product Support

M. Votava, W. Bliss, S. Cutts-Bone, C. Debaun, F. Donno-Raffaelli, R. Herber, K. Leininger, B. Lindgren, J. Nicholls, G. Oleynik, D. Petravick, R. Pordes, L. Sexton, J. Streets, B. Troemel, L. Udumula, and M. Wicks

Access Liaison, CDF, Distributed Computing and Online Support Departments

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

June 1991

* Presented at the *7th Real Time Conference*, June 25-28, 1991, Julich, Federal Republic of Germany.

UPS UNIX Product Support

Margaret Votava, William Bliss, Sandra Cutts-Bone, Chuck Debaun, Flavia Donno-Raffaelli, Randy Herber, Kevin Leininger, Bert Lindgren, Judy Nicholls, Gene Oleynik, Don Petravick, Ruth Pordes, Liz Sexton, Jonathan Streets, Brian Troemel, Lourdu Udumula, Matt Wicks

Access Liaison, CDF, Distributed Computing, and Online Support Departments
Fermi National Accelerator Laboratory
Batavia, Illinois 60510

Abstract

The need to provide central support and distribution of many software packages across a variety of UNIX™ platforms at Fermilab has led to development of a methodology, UPS, for the packaging, maintenance, and distribution of our software. UPS has now been implemented and in use for almost a year on four different UNIX platforms. This paper discusses the goals of the software, implementation of the product, and experiences in its use.

I. INTRODUCTION

Four different UNIX platforms are given support by the Fermilab Computing Division. The need for central support has led to the design and implementation of a methods to manage software. The primary purpose is to facilitate the development, maintenance, distribution, and access of the software for which the Computing Division is responsible.

This methodology, UNIX Product Support (UPS), is in use on SunOS™ , IRIX™ , ULTRIX™ , and AIX™ systems at Fermilab. Working within the UPS framework, developers follow a recommended structure for software products with various tools to support this structure. Users have a uniform access to versions of software. A distribution procedure allows remote users to obtain information about and acquire available software while allowing the maintainers to monitor and log these accesses.

UPS exists as two distinct parts,

- A database that contains information about each product and its instances
- A set of procedures and/or programs to manipulate the database.

UPS provides a central place for keeping information about the products available on a UNIX system, while allowing the product files themselves to be on different disks and in different directories. Additionally, UPS allows for different *instances* of a product to be available on a system. A product instance is a unique combination of product name, version, and flavor. Flavor is an arbitrary string that can help distinguish a particular version. For example, v1.0 may have two instances in the database: one with a flavor of AIX and one with ULTRIX to reflect on which platform it runs.

II. DESIGN GOALS

UPS was designed to address the needs of users, supporters, and developers of UNIX software. It has the following features:

- Simple interface for the typical user.
- Minimal requirements for packaging existing software for access through UPS.
- Support for multiple groupings of software products for use and maintenance by independent working groups (e.g., experiments).
- Ease of porting between UNIX platforms.
- Support for a well structured development environment.
- Support for multiple versions of a product.
- Support for platform specific versions of products.
- Structure for actions to be performed at system boot and shutdown time.
- Support for dependency between requirements.

III. UPS COMMANDS

UPS commands can be run by anybody but typically different classes of users are interested in different subsets of commands. All commands have a similar set of switches to alter default behavior.

™ UNIX is a registered trademark of UNIX Systems Laboratories
™ SunOS is a registered trademark of Sun MicroSystems
™ IRIX is a registered trademark of Silicon Graphics, Inc.
™ ULTRIX is a resitered trademark of Digital Equipment Corporation
™ AIX is registered trademark of International Business Machines

User Commands

setup	Make a product available for general use. Would typically execute set any aliases and environment variables necessary to use the product. Lasts for the duration of the process.
unsetup	Make a product unavailable. Typically unsets aliases and environment variables.
ups list	List all the products available.

Product Developer Commands

ups declare	Make a product instance known to the database.
ups undeclare	Remove a product instance from the database.
ups modify	Modify an entry in the database.

Product Installer Commands

ups tailor	Tailors a product instance. Typically used for products on clustered environments that only do something special on one node, e.g., drivers.
ups start	Start a product instance. Typically used for starting products that run daemons or cron jobs.
ups stop	Stop a product instance. Typically would stop any processes started by ups start.

IV. IMPLEMENTATION

To maximize portability and speed, the UPS database is a set of ASCII text files and the manipulation routines are written in C. UPS currently operates in the cshell environment, although has been designed to work under any shell. Figure 1 shows the breakup of ups functions into various components. Shaded bubbles indicate functionality not yet available.

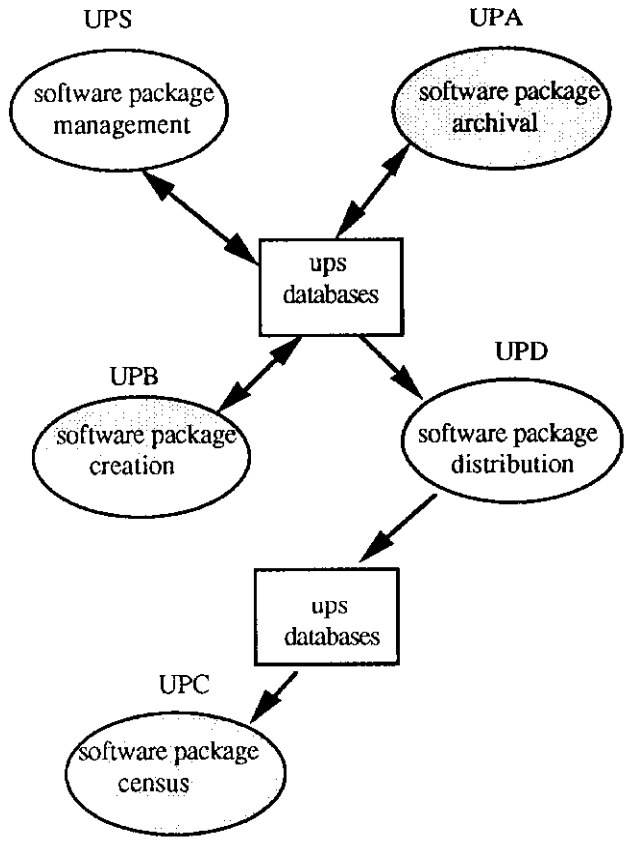

Figure 1. UPS Family of Products

UPS Database

The UPS database is not a discrete file that contains information about all products on a system; instead it is directory tree. Each individual product has an ASCII file inside the database directory that describes itself and all of the available instances (e.g., version name, flavor, date declared, etc). This directory can be named anything on any mounted file system. UPS commands find this directory via the environment variable PRODUCTS. PRODUCTS can be set to point to several directories thus giving support for multiple databases.

UPS Product Structure

The required directory structure and file organization for a product to be supported by UPS are minimal. A product instance need simply lie underneath a single root. If additional UPS functionality is desired, all special UPS files must live within a top level subdirectory called ups; however, none of these special files are required.

The following special files are currently supported: **setup**, **unsetup**, **current**, **uncurrent**, **start**, **stop**, **tailor**. **setup** is executed when a product is setup. It typically defines any necessary aliases and environment variables needed to use the product. The **unsetup** file is run when the product

instance is unsetup. **current** is executed when the product instance is declared current. It typically moves any necessary executables to usr/local/bin. **uncurrent** is run when the product is no longer current. **start, stop** and **tailor** are run at start, stop and tailor time, respectively. All files except setup and unsetup can either be binaries or shell scripts. Setup and unsetup must be shell scripts.

Two subdirectories within the UPS subdirectory are also supported. Any file in the **toman** directory will be copied into the system's **man** page area. Any file in the **toInfo** directory will be copied to the system's **Info** area.

V. Distributing UPS Products to Remote Sites

UNIX Product Distribution (UPD) is a companion product to UPS. It allows the distribution of software products, stored as **tar** files, from a host to a remote node. UPD provides commands to list available products, list the contents of a **tar** file, copy a product or copy an individual file from a product **tar** file.

Once on the local node, UPD can optionally be used to unwind the **tar** file and delcare the product to UPS. UPD supprts three user interfaces: a command line mode, a simple menu interface and an X Windows interface. The X Windows interface provides a user-fiendly way of listing the files in a products and extracting useful files such as documentation, to a local node. In order to allow distribution of proprietary software, UPD can encrypt a **tar** file when creating it on the host side. A local site can copy it, but will not be able to unwind it unless given the decryption key.

On the host side, UPD is implemented using a captured account, which enables monitoring and logging of distributions. UPD supports user selection of the host database to be used.

VI. Future Enhancements

Several enhancements to UPS are in development.

UPC	UNIX Product Census. UPC will provide a mechanism to query remote sites about their product databases. It will compile product summaries and email them to the appropriate users. It can be submitted to cron to run on a scheduled basis. UPC will enable product maintainers to make sure various remote sites have the current releases of software.
UPA	UNIX Product Archive. UPA will provide a mechanism of copying products in UPS format to magnetic tape for archival purposes. It will also provide for the retrieval of these products.
UPB	UNIX Product Build. UPB will build a distributable version of a product instance from the development version. This will provide a uniform method of creating "cut" versions. It will also support building of the same version on multiple UNIX platforms.

VII. Experiences Using UPS

The development of UPS was a very broad effort with participation from various groups of the Computing Division as well as several experiments. UPS has been available in various levels of completion at Fermilab since August, 1990. It is in widespread use both on the centrally supported UNIX development systems, offline computing farms, and experiment UNIX workstations.

A major Fermilab experiment, CDF, uses UPS and UPD to manage, maintain, and distributed their complete experiment offline analysis code. About ten remote sites actively distribute software from the CDF central support nodes. The Computing Division maintains a central distribution host on the main Silicon Graphics development system and on site and off site users can register to distribute software. Figure 2 shows the configuration of the central distribution machine.

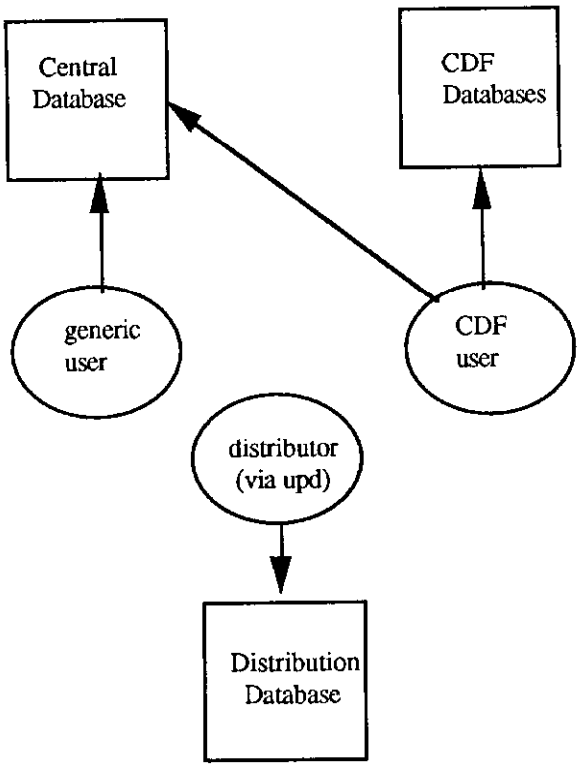

Figure 2. Central Facilities Configuration

Several groups take advantage of UPS' capability of supporting multiple product databases on one computer. In the Online Support Department we use a development UNIX LAN consisting of four different UNIX platform types of UNIX workstations operating from a single UPS database. Figure 3 shows a configuration that supports multiple UNIX platforms. Product **unix_hoist** has executables for all three platforms. It is registered three times in the database, once for each platform. UPS "knows" which binaries to run on on which machine.

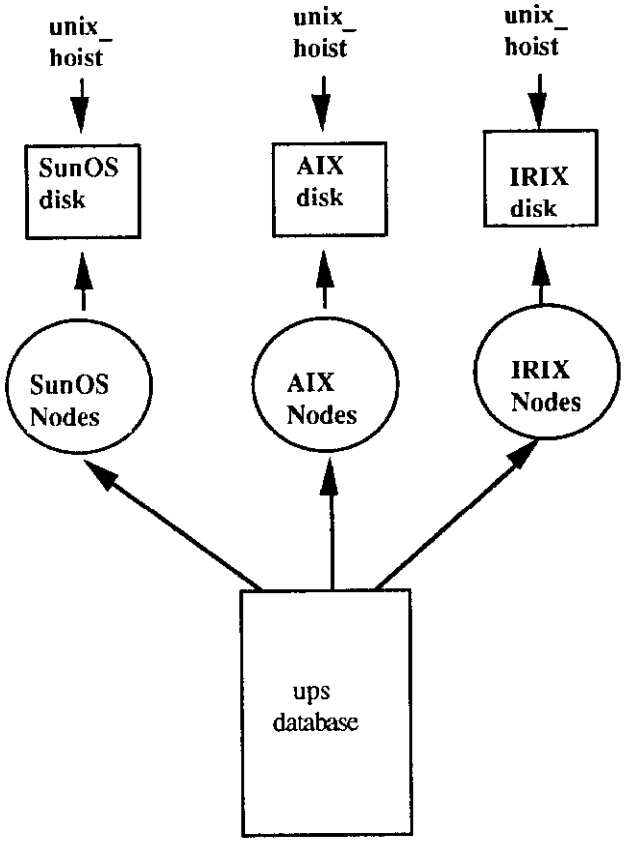

Figure 3. Multiple Platforms Sharing a Single Database

UPS and UPD are now part of the *Fermi UNIX Environment*. The development and availability of UPS and UPD early in the life time of UNIX at Fermilab has been of great benefit in establishing an organized approach to the packaging and support of the many products needed by experimenters. The minimal requirement imposed by UPS have enabled us to easily package and distribute commercial software. UPS has delivered its parcels of UNIX goodies to the physicists' doorstep.

VIII. REFERENCES

[1] C. DeBaun, et al, *UNIX Product Support Design and Implementation*, Fermilab internal document, PN425.

[2] K. Leininger, et al, *UPS Start/Stop Design*, Fermilab internal document, DS216.

[3] M. Votava, et al, *UPS User's Guide*, Fermilab internal document, PN426.

[4] J. Streets, et al, *UNIX Product Distribute User's Guide*, Fermilab internal document, PN435.

[5] M. Wicks, *Fermi Unix Environment DCD Products User Guide*, Fermilab internal document, DCD3.2.

[6] J. Nicholls, *Fermi UNIX Environment*, presented at Computing and High Energy Physics Conference, KEK, Tsukuba, Japan, March 10-15, 1991..

[7] L. Udumula, et al, *UPS Product Census*, Fermilab internal document, DS208.

[8] S. Cutts-Bone, *Overview of Product Installation and Distribution on UNIX Machines*, Fermilab internal document, GU0006.