Chapter 20: Checklist for Building and Distributing Products

In this chapter we summarize the steps for preparing to build a product, building it and distributing it. We include information about making the appropriate announcements when a new or upgraded product is available.

20.1 Pre-build Checklist

1) Create product root directory structure. Here is a comprehensive list of product elements and their suggested subdirectories (most products don't require all of them):
   - `README` (top-level) and `RELEASE_NOTES` files (top-level or `ups`)
   - `INSTALL_NOTE` file (`ups`)
   - top-level Makefile
   - executables (`bin`)
   - table file and other installation-independent files/scripts (`ups`)
   - source code and build instructions (`src`)
   - Makefile for build (`src`)
   - html user documentation (`html`)
   - PostScript or text user documentation (`doc`)
   - unformatted man pages (`ups/toman/man`)
   - formatted man pages (`ups/toman/catman`)
   - test scripts (`test`)
   - examples (`examples`)
   - libraries (`lib`)
   - include files (`include`)

   If you use **template_product**, the operation of cloning it creates the product root directory, the top-level file templates and Makefile, several of the listed subdirectories, and a basic table file.

2) For shell script or pre-built binary products, put the executable file(s) in

the `${UPS_PROD_DIR}/bin` directory.

For products requiring build, create the file `${UPS_PROD_DIR}/src/Makefile`. (Include instructions for compiling, linking, testing and all other necessary operations, as well as for copying the final binaries into `${UPS_PROD_DIR}/bin`.) Insert the product source code into `${UPS_PROD_DIR}/src`.

3) Include documentation (html, man pages, user guide).

4) Create/edit `README` (and `INSTALL_NOTE` and `RELEASE_NOTES` as needed). See samples in sections 17.3.1 *README*, 17.3.2 *INSTALL_NOTE* and 17.3.3 *RELEASE_NOTES*. **template_product** creates template files that you need to edit.

5) Create/edit the table file (usually under `${UPS_PROD_DIR}/ups`). See section 36.7 *Table File Examples*. **template_product** creates a basic one that you need to edit.

6) Create any extra scripts your product needs in `${UPS_PROD_DIR}/ups`. See Chapter 37: *Scripts You May Need to Provide with a Product* for examples.

7) Create/edit the top-level Makefile (include targets for building the product, setting permissions, testing, distributing, and so on). Section 19.4 *The Top-Level Makefile* lists the first part of the Makefile that comes with **template_product**, for reference.

8) (Optional) Declare the product to a local database (use the **`-d`** flag).

9) Store the master source code and all the auxiliary files in a **CVS** code repository (or other code-version management system) according to your group's policies.

For OSS group:
`CVSROOT=cvsuser@cdcvs.fnal.gov:/cvs/cd`.

20.2 Build the Product

10) Verify that dependencies required for build are present.

11) Build the product using `${UPS_PROD_DIR}/src/Makefile` (should get called by top-level Makefile).

12) Set permissions to **`a+x`** for scripts and other executables, and to **`a+r`** for readable files (should get done by top-level Makefile).

13) If using **template_product**, modify the top-level Makefile to include

build instructions and other targets, as needed, and use the top-level Makefile for subsequent builds.

20.3 Test the Product

14) Declare the product to a local database, if you haven't already.

15) Verify that dependencies are present.

16) Run **`ups verify`** on the product to check the integrity of the database files (this command is described in section 23.20 *ups verify*).

17) Setup and test the product (test scripts should get run by top-level Makefile).

20.4 Distribute to fnkits as "test"

18) Make sure you're registered to add products to *fnkits*. (Send email to *compdiv@fnal* to request authorization.)

19) If product should have special access restrictions, fill out the **Special UPD Product Registration** form (at `http://fnkits.fnal.gov/specialprod.html`).

20) (Optional) Make a tar file of your product.

21) Add the product to *fnkits* as "test".

If using **template_product**, run **`make kits`** from the product root directory (it sets the chain to whatever CHAIN is set to in the Makefile). Otherwise use **`upd addproduct`** (should be called from your top-level Makefile). Here is a sample **`upd addproduct`** command:

```
% upd addproduct       \
<product> [<version>]  \      # product name and version
-t                     \      # "test" chain
[-f <flavor>]          \      # flavor
[-q <qualifierList>    \      # qualifiers
[-T <tarFile>]         \      # path to tar file
[-m <tableFile>]       \      # table file name
```

20.5 Announce the Product

22) Make documentation available on-line under `http://www.fnal.gov/docs/products/<product_name>` (`/afs/fnal.gov/files/docs/products/<product_name>`). Include html documentation.

23) Fill out the on-line **Computing Division Product Input Form** at `http://cddocs.fnal.gov/cfdocs/productsDB/productinput.html` to inform the products database maintainer about your product arriving on *fnkits*.

24) If appropriate, install the product from *fnkits* onto *fnalu* as "test".

25) Post news to `fnal.announce.products`, `fnal.announce.unix` (if it is a UNIX product), `fnal.sys.fnalu.announce` (if installed on *fnalu*).

26) Create <*product*>*-users@fnal.gov* mailing list (if appropriate), and send email announcing test phase.

20.6 Distribute to fnkits as "current"

27) Wait suitable time (amount of time depends on product).

28) Fix problems found during test phase.

29) Rebuild product.

30) Commit changes to **CVS**.

31) Put final release into *fnkits* as "current".

32) Reinstall as "current" on *fnalu*, as appropriate.

33) Check all the chains on *fnkits* (and *fnalu*) to make sure that older versions, flavors, etc. are no longer chained to "current".

34) Post news to *fnal.announce.products*, *fnal.announce.unix* (if it is a UNIX product), and *fnal.sys.fnalu.announce* (if installed on *fnalu*).

35) Send email to <*product*>*-users@fnal.gov* announcing current phase.

36) Send email to *helpdesk@fnal.gov* to inform them about the new product or version. Include information on the kinds of questions to expect, if possible, and where to direct users for help.