

An experimental CTAN upload process

Jim Hefferon

ftpmaint (at) tug dot ctan dot org

Abstract

Some experimental software may improve the way in which packages are handled at the Comprehensive T_EX Archive Network (CTAN).

1 Now

CTAN is run at three different sites, one in Germany, one in Britain, and one in the US. Any adding, deleting, or moving of files happens at one of these three. A custom program, written by Rainer Schöpf, ensures that a change at one is quickly reflected at the other two, within fifteen minutes. The more than one hundred other CTAN mirrors go at a different pace, usually syncing nightly.

New or updated material reaches us in three ways. Usually it sent by an author via a web form. Besides that, some authors send it via FTP, and some packages are automatically mirrored in from other sites. The author-sent cases could be either new packages or updates, while the automatic case only applies to updates. I will focus on the web uploads.

In the present system, a web upload triggers an email to the CTAN maintainers mailing list. The maintainer at the site receiving the material sees the email and handles the upload. This means unpacking the `.zip` or `.tar.gz` bundle in which the files were sent and examining the resulting files to check details such as license and placement. It may mean writing to the author or to the other maintainers, for instance to ask the author for documentation. After that, the maintainer runs Rainer's program to put the material into the archive and trigger the mirroring by the other core sites, and so ultimately by the additional mirrors.

Placing a package's files into the archive does not end its processing. Information about the package such as description and license—the package's metadata—needs to go into the *Catalogue*; this is done by Robin Fairbairns. Finally, distributions such as MiK_TE_X and T_EX Live repackage the material to meet the T_EX Directory Standard (TDS) and deliver it in this convenient form to typical end users.

The process above has some advantages. In particular, at an archive such as SourceForge where responsibility for how a package is offered lies with

the author, some percentage of the authors do not do a good job. But at CTAN the maintainers see that packages meet some standards. So a current strength of CTAN is that it is a wide-mouthed funnel, catching a range of submissions and narrowing them to a more uniform offering.

However, no doubt the process could be better. Here are a few concerns that we have heard.

1. Authors cannot conveniently edit the metadata.
2. There are delays of various kinds. One example is that package metadata often gets into the database only after the files are in the archive, so there is a period where the description does not match the package. Another example is that the web pages for the archive at <http://www.ctan.org/tex-archive> are usually regenerated nightly, so information about new materials is not current.
3. To be a core maintainer a person needs to run a server and there are people who could help with the archive but who oughtn't administer a system that is exposed to the Internet (including me).
4. The package gets installed by the maintainer whose site happened to receive the upload, so if that person is unavailable then there is a wait.
5. Many of the steps are done by hand, which can lead to errors.
6. At the time that a package is put in the archive and announced, it should be convenient for end users to install.

2 Developments

Users groups, notably Dante, have sponsored very helpful discussions of CTAN issues. In response, I have been working on software that is now being deployed and tested. The upload process described here still faces a fair number of hurdles. But some people have expressed interest and it is in an advanced enough state that the outline below may help these folks to get a rough understanding of what it does.

If you are not keen on CTAN internals then probably the feature that is the most interesting to you is also the most experimental. The T_EX Live team has a script to bring most packages from the CTAN tree over to the standard T_EX Directory Structure layout, that is, over to a format that could be dropped by an end user into their existing installation. The process described here wraps that script to make the TDS-ready material available as a $\langle package\ id \rangle.tds.zip$ bundle at the time that the package is put into the main CTAN archive. This is a regular ZIP file and users can `unzip` it right into their distribution tree, without much need for instructions. (This does not integrate with any package manager but it does allow users to easily place material that they want.)

To describe the process I will walk through the steps that a typical package would take to get from author to archive.

1. The author puts the package into a `.zip` or `.tar.gz` bundle. They visit CTAN's upload web address and first select whether the upload is a new package or an update of an existing package.

They then see the main upload form. Probably they fill out the simple version that asks only for name, license, and description. But more adventurous authors can get a form to specify more obscure attributes, such as the package home page.

If this is an update of an existing package then when the form appears it already has the metadata that is now in the database and the author just makes any changes. The author is asked separately for additional information such as any handling instructions (in the current system, the description and handling information goes in the same input box).

2. The system accepts the uploaded package and metadata. It places the metadata in the database, in a pool of not-yet-processed uploads. It sends an email to a list of people who can edit and install uploads, called here "editors".
3. The contributor's uploaded bundle is unpacked to a file tree by a program that runs periodically. (This does not happen as part of accepting the upload because the author's bundle must be unpacked in a secure way, in a chroot jail.)

This program does a few things beyond unpacking such as resolving text file line endings issues. When it finishes, it sends a notice to the email list of upload editors.

4. One of the editors sees the notification and logs into a web site listing the pool. They have a peek to see if the material is something that they could handle right now and if so then they claim responsibility for it.
5. This editor examines, possibly edits, and then approves the metadata left by the contributor. (Requiring that metadata be approved reassures authors that people they don't know cannot change the package's description.)

The editor can read, add, delete, or rename files. For instance, they can delete a `.svn` file that was accidentally included in the upload.

This page warns the editor if there are some problems. One example is that a warning will appear if the metadata says a `README` file exists but there is not one in the uploaded file tree. Another example is that a warning appears if an install will leave soft links dangling in the archive.

6. The author may have included in the upload their own $\langle package\ id \rangle.tds.zip$ file. If so, the editor can see its contents and compare with what T_EX Live now has for this package, if anything.

The editor can also push a button to make a new `.tds.zip` bundle, using T_EX Live's script. If the package is suitable for T_EX Live (which in most cases means only that it satisfies the license restrictions) then it can be placed in the local Subversion sandbox for later commitment to the T_EX Live repository. In either case, if the T_EX Live script does not succeed then the page makes that obvious.

7. The editor then pushes a button to install the material.

That puts the source files to the archive, say at `/macros/latex/contrib/⟨package id⟩`. Installation is done using the metadata so the database and the archive tree are consistent regarding the location, whether a `.zip` file exists of the directory contents, etc. Files are placed with Rainer's program, ensuring that these web-based installations are consistent with command-line installations.

The installation system also tends the database: it updates the metadata and the searchable documentation.

If there is a TDS bundle then the system puts it at a place related to where the source files went, such as `/install/macros/latex/contrib/⟨package id⟩.tds.zip`.

8. The installation routine sends an email to the editors list, telling people that the upload has

been handled, and for possible forwarding to the CTAN announcement mailing list.

9. If a `.tds.zip` bundle was queued in the local T_EX Live Subversion sandbox then the system will periodically try to commit the changes to the T_EX Live repository. One advantage of doing this at a separate time than the moment of installation is to guard against network connectivity problems between the CTAN site and the T_EX Live site. Another advantage is that when the T_EX Live folks are getting ready for a new release then this job can be shut off.

Material that comes in as an FTP upload goes through the same process, starting at step 3 (there is a way to associate metadata with the upload). This system has no way to handle materials that arrive automatically.

3 To do

Not every feature of the experimental system is described above; for instance, there is a way for authors to send changes to the metadata alone. And, because it is experimental, probably some of what is above will be changed if it ever reaches a production status. In particular, while the TDS feature appears promising, it is quite experimental.

So the upload process described here still faces a fair number of hurdles, both technical and non-technical. For one thing, where the current upload process is like a wide-mouthed funnel, the process described above has not been subject to any real-world testing for the same property. However, all the features described above exist, are now being developed and tested, and seem to solve at least some of the problems with the current package process.

4 Acknowledgement

Thank you to Karl Berry for help with the T_EX Live material.