

CCP4 Package Manager

Eugene Krissinel and Andrey Lebedev

CCP4, Research Complex at Harwell, Rutherford Appleton Laboratory, Didcot OX11 0FA, UK

Starting from release 6.3.0, CCP4 introduces a new tool, CCP4 Package Manager, in the Suite. This tool facilitates installation of precompiled CCP4 Suite on Linux and Mac OSX platforms, and provides updates to the Suite on all systems. For technical reasons, Package Manager is split in two application: *ccp4sm* (setup manager) and *ccp4um* (update manager), which, however, are compiled from the same source code.

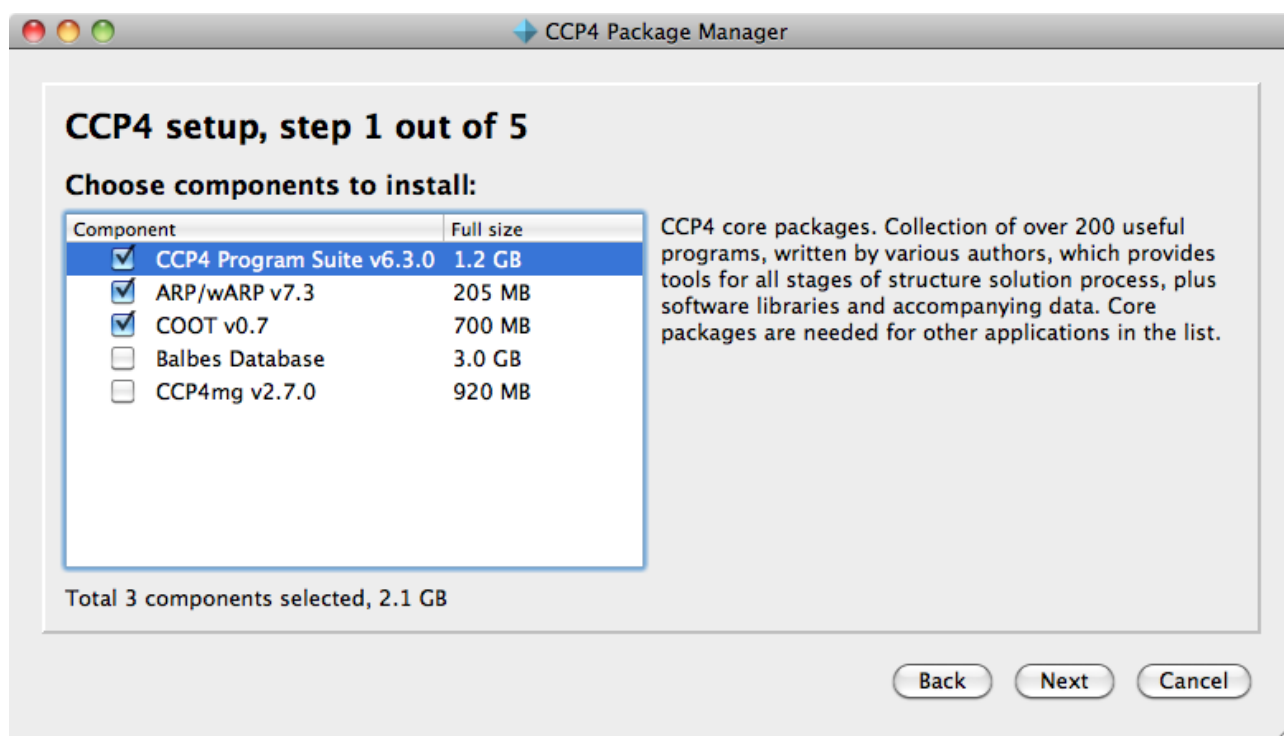
CCP4 Setup Manager

CCP4 Setup Manager, *ccp4sm*, represents a GUI front-end for the installation of CCP4 Software Suite. It does nothing else but exactly the same actions as a user would do in manual mode: identifying desired packages (CCP4 Core, Arp/wArp, Coot, Balbes database, CCP4 MG), downloading the corresponding archives (tarballs on Linuxes, or dmg images on Mac OSX platforms), unpacking, configuring and running installation scripts. Using *ccp4sm* gives extra convenience on comparison with installation in manual mode:

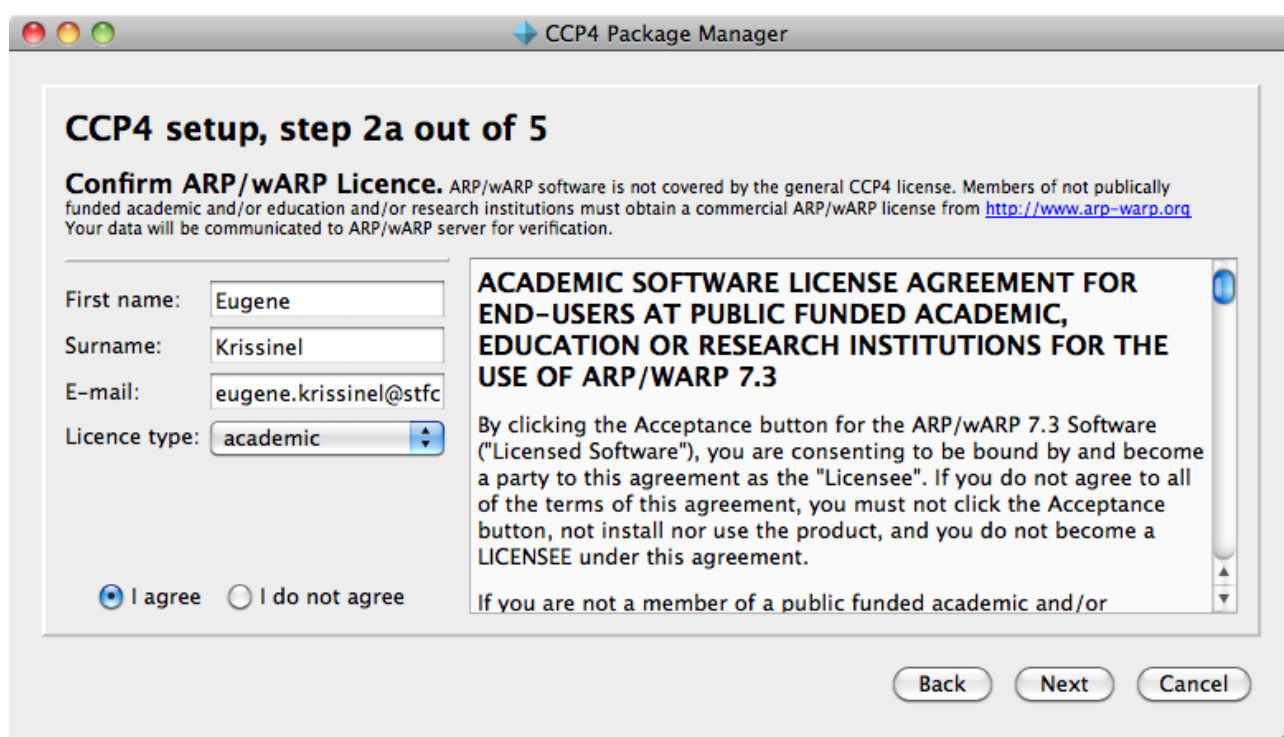
- decreased chances to install pre-compiled binaries incompatible with user's platform (*ccp4sm* will still allow, with a warning, installation of 32-bit packages on a 64-bit machine, if the corresponding libraries are available)
- download all required packages as a single action
- provide a resumable download, useful where network connection is poor
- will match specific packages to user's platform where pre-compiled binaries are not fully portable (such as Coot and, to the extent, CCP4 MG)
- does not require editing of setup files and working in Terminal
- cleans up temporary files and puts application icons (ccp4i, Coot, MG) on Linux Desktop

Before using, *ccp4sm* needs to be downloaded from CCP4 web site. It represents a light-weight graphical application (about 7MB single file on Linux platforms and a 15MB dmg package on Mac OSX), which will launch after double-clicking on it in OS's file manager (such as Nautilus or Finder). Full installation process is done in 5 steps, and is presented in 5 pages in *ccp4sm*, which are navigable back-and-forth using "Back" and "Next" buttons. At any step, installation may be cancelled by pushing the "Cancel" button.

Right after starting, *ccp4sm* establishes connection with CCP4 server. Then, as the 1st installation step, the user is presented with the choice of packages for installation:



After package selection, on the 2nd step, the user is prompted to confirm agreement with all applicable licences. In case ARP/wARP is included in the list, *ccp4sm* also queries ARP/wARP server in EMBL-Hamburg for authorisation. Certain user data: name and a valid e-mail address, is collected for this process:



Commercial users also need to specify their licence number, which should be acquired from EMBL-EM prior installation. Authorisation is usually instantaneous if network connection is stable.

On 3rd step, the user is asked about particular location (destination directory), where selected packages should be installed, as well as directory for temporary files. On Mac OSX platforms,

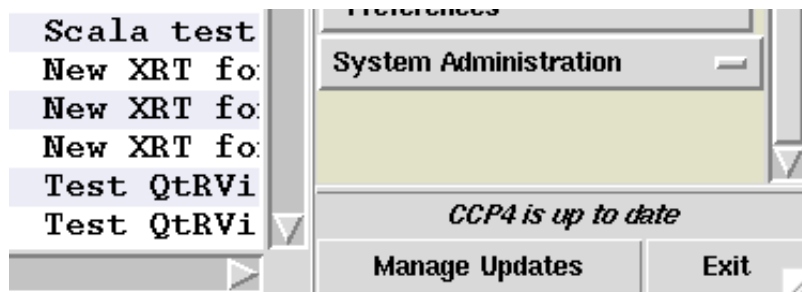
installation is limited to `/Application` folder, and `/tmp` directory is used for storing all intermediate data files. On Linux platforms, `ccp4sm` checks whether the user has sufficient privileges for writing into the destination directory, and requires a restart in administrative mode if user privileges are found insufficient.

Step 4 includes download of all selected packages. This may be a relatively long process, varying between 10-20 minutes up to a few hours, depending on the network speed, during which no further actions are required from the user. At any moment, the download may be stopped and resumed later, even if `ccp4sm` is restarted after a lengthy time period. If network connection is lost, `ccp4sm` will resume download from last recorded position, when it is started next time.

On the last step, `ccp4sm` runs installation scripts, which unpack downloaded files, put them into required location, configure and execute post-installation procedures where required. Upon completion, `ccp4sm` puts icons on user's desktop (Linux only), writes installation log `install.log` into destination directory, and removes all temporary files. After `ccp4sm` finishes, CCP4 packages are ready to use.

CCP4 Update Manager

CCP4 Update Manager, `ccp4um`, was introduced as an add-on feature shortly after CCP4 Release 6.3.0 in July 2012. In order to install `ccp4um`, a user would need to download a pre-compiled binary (Windows and Linux) or a dmg image with *.app folder (Mac OSX), copy it into `$CCP4` location and run by double-clicking in system's file manager. At first run, the updater integrated itself into the suite by creating `$CCP4/libexec` and `$CCP4/restore` directories and setting up the "Manage Updates" button and update status line in `ccp4i`:



In December 2012, CCP4 Release 6.3.0 was re-bundled to include `ccp4um` as a standard feature, eliminating the need to install it separately.

CCP4 Update Manager represents a graphical client application, which modifies CCP4 setups by downloading patches from CCP4 server and applying them locally. A "patch" may be either a file or an action to be applied to existing files. In update mechanism for CCP4 release 6.3.0, the following operations are supported:

- replacing a file
- placing a new file
- deleting a file
- changing file permissions

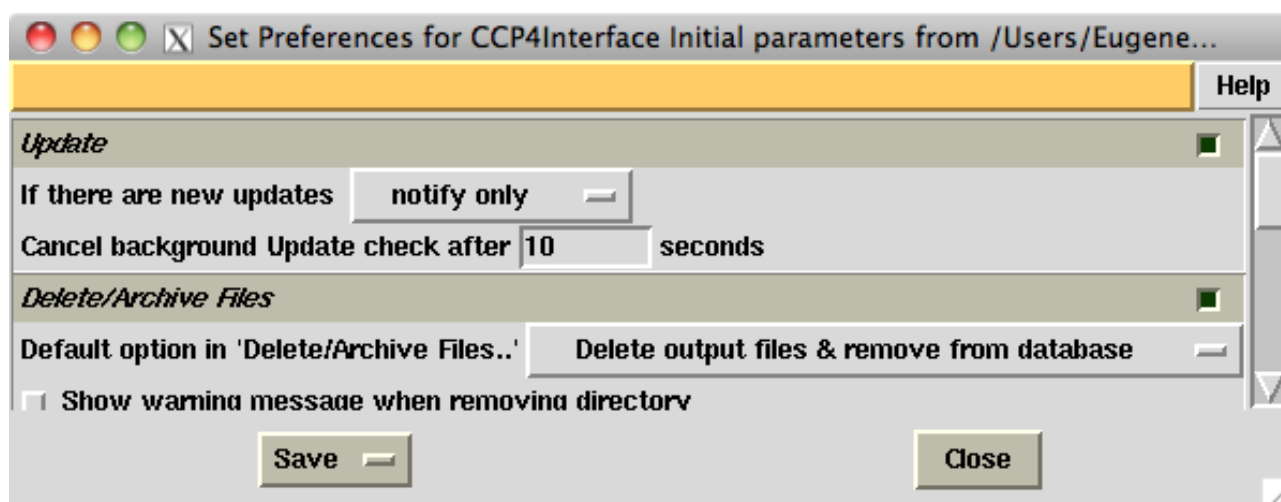
Starting from Release 6.4.0, they will be complemented with

- replacing/making/deleting a soft link
- running a process

the need of which has been identified from the experience with 6.3.0 update series.

Before applying updates, the updater makes an incremental backup of CCP4 setup, which is stored in `$CCP4/restore` directory. This allows a user to roll back, if any particular update was found harmful or unsuccessful. As an essential feature of CCP4 updates, they may be applied only consecutively, and rolled back strictly in reverse order. This restriction is imposed by the updater automatically. Even if most updates do not interfere with each other, tracking all possible dependencies in a relatively large Suite is difficult, therefore, only a single line of Suite evolution on client machines is allowed.

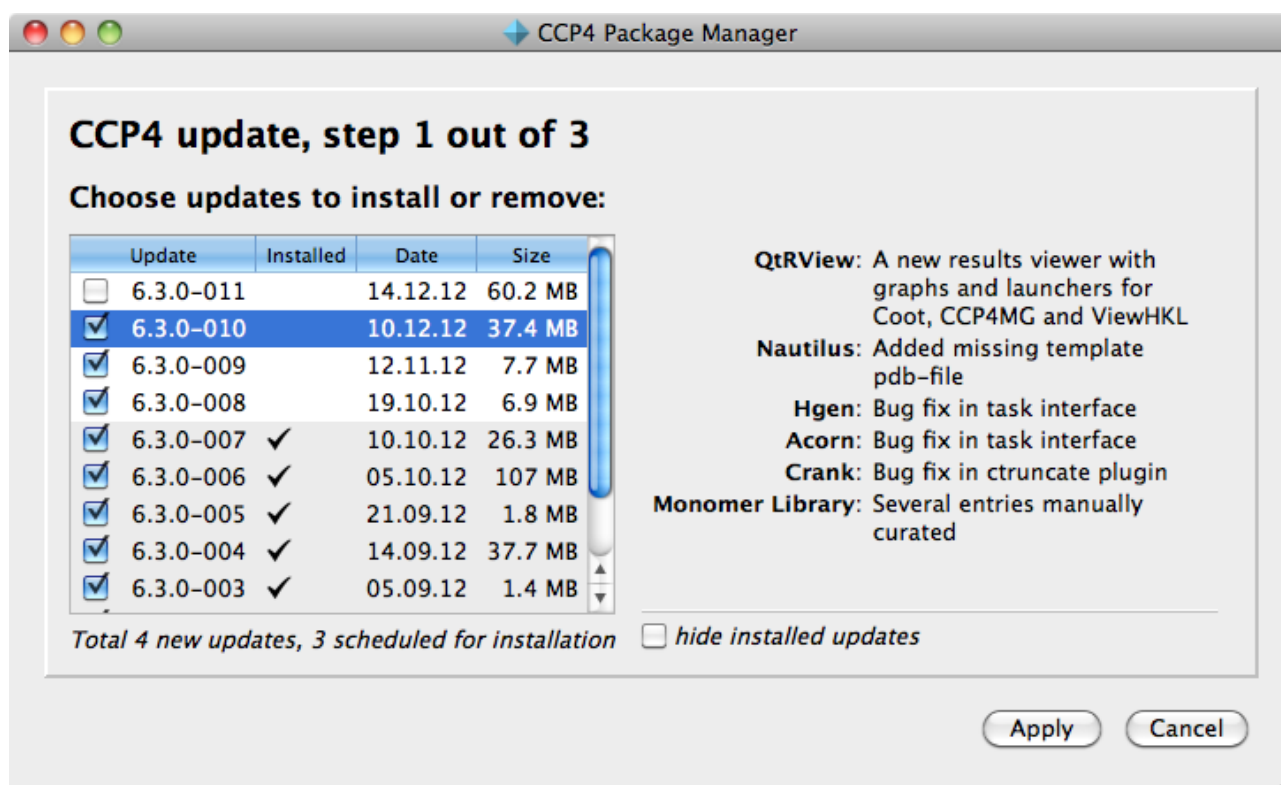
The updater may be started either from *ccp4i* (by pressing the “Manage Updates” button), or from command prompt by running `$CCP4/bin/ccp4um`. When *ccp4i* starts, it runs *ccp4um* in silent mode to check whether new updates are available. This process is controlled by *ccp4i*'s Preferences, where one of three update modes may be selected:



The three modes are:

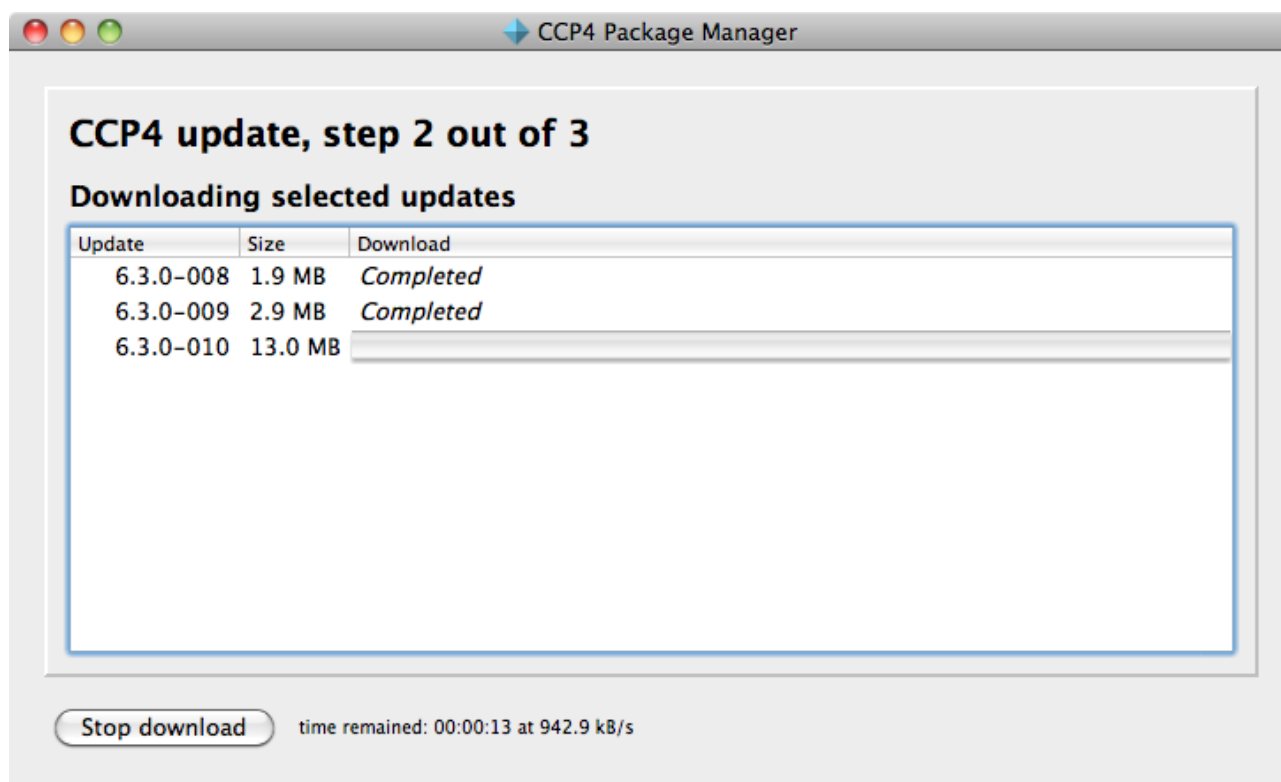
- do nothing: no checks for new updates are done
- notify only. This is the default mode, in which *ccp4um* runs silently each time *ccp4i* is started. If *ccp4um* identifies new updates on CCP4 server, a message “There are N new updates” is displayed in update status line in *ccp4i*, just above the “Manage Updates” button. At this point, the user may start *ccp4um* by pushing the “Manage Updates” button and install the updates
- start updater. In this mode, *ccp4i* acts same way as in “notify only”, however, in difference, *ccp4um* is started automatically in full interactive mode if silent check finds that new updates are available.

After starting in full graphical mode, *ccp4um* presents the user with the list of installed and available updates:

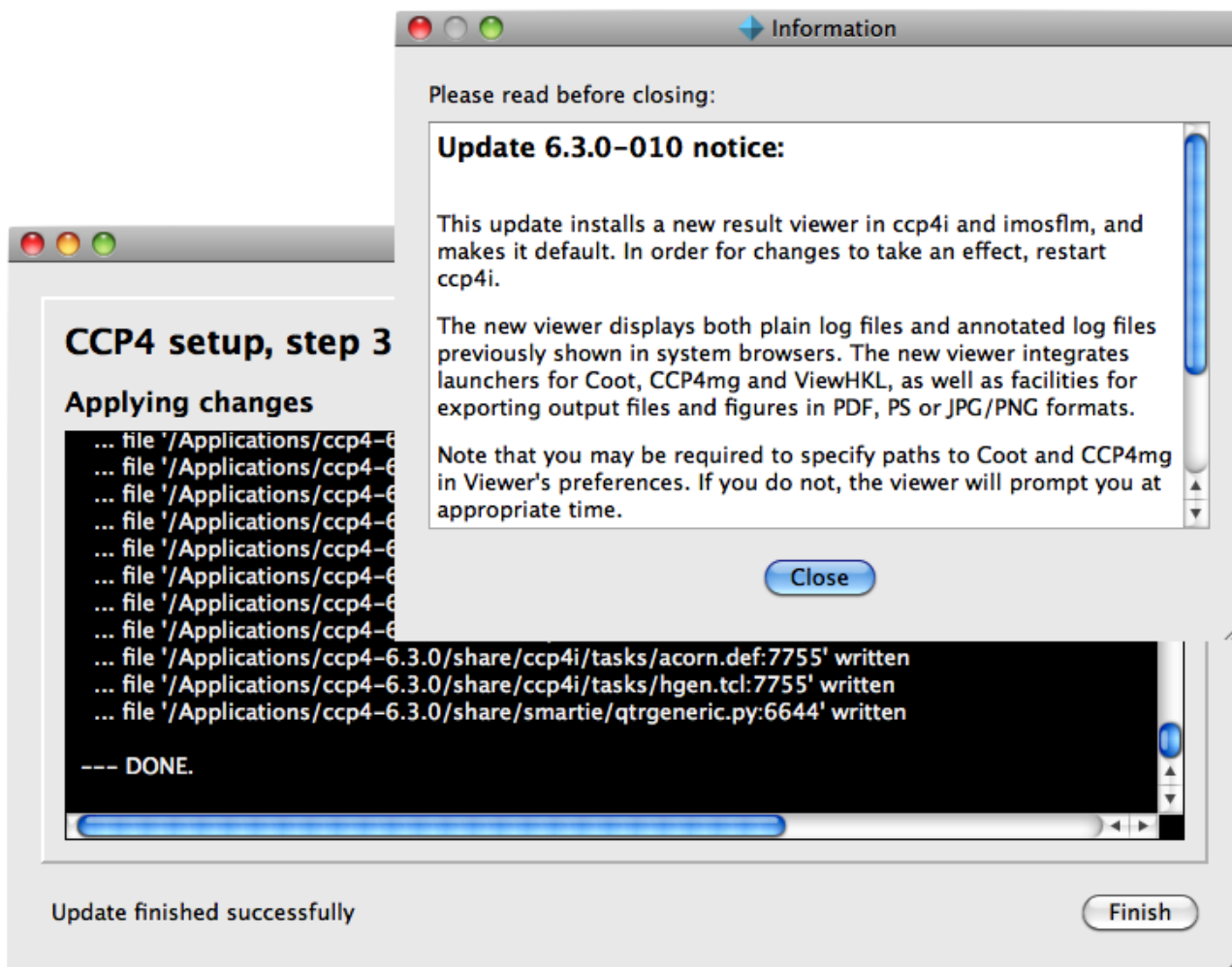


Selection of a particular update in the list will show a brief description of the update's content. In order to install update(s), a user needs to check the corresponding box(es) in the left-most column of the list, and press the "Apply" button. For uninstalling update(s), the corresponding box(es) need to be unchecked.

After pressing the "Apply" button, the rest of the update process goes automatically. First, the updates are downloaded from CCP4 server:



After downloading, *ccp4um* makes an incremental backup of changes to-be-made, and, finally, applies all patches. Optionally, *ccp4um* may display a pop-up window to advise the user of any unconventional effects the update may cause, or to re-start *ccp4i* if required:



The server-side of CCP4 update mechanism represents a set of tools for identifying updated components, packing them in archive files and placing in a designated location on the server. For this purpose, a set of template CCP4 setups (one per supported OS), representing the current state of the Suite on client machines, is maintained. Once updated component(s) are placed into the template setup, a special procedure calculates the induced difference, advances the update counter and generates the update description files (UDF). These files are retained for bookkeeping purposes and may be used to re-generate updates at any time. After generation, UDFs need some manual editing in order to add the update's annotation and optional pop-up messages to be displayed by *ccp4um*. After that, a separate process uses thus prepared UDFs to produce update packages and places them on the server. Update packages represent compressed archives containing the updated components, their descriptions and other service information for *ccp4um*.

Summary of all updates is available at <http://www.ccp4.ac.uk/updates/>, together with cumulative update patches. These patches may be used by those users who cannot run *ccp4um* because of, e.g., restricted network access.

During first 8 months of *ccp4um* deployment, CCP4 core team has issued 21 updates in 6.3.0 series. These updates contained mostly bug fixes, but also new versions of important components, such as Phaser 2.5.2, released in Update 6.3.0-007 in October 2012, as well as all-new applications, such as the new result viewer, QtRView (Update 6.3.0-010 in December 2012). It is

expected that, in future, most changes in the Suite will be delivered in update mode, rather than by means of full-scale releases, which, therefore, will become less frequent. The logics behind having both updates and releases is the same as that of virtually any other complex software, which normally has update and upgrade cycles. While updates are extremely efficient for delivering relatively small and local changes in the Suite, full releases are necessary when essential structural changes or new underlying technologies need to be implemented.