

## **Update Nag**

- The data for the update nag is kept in text files of sql commands and the servlet properties file. There are two tables, versions and policies.
- The code for the update nag servlet can be found in updates.java
- The update nag is used to notify users when we have a new beta / release.
- The client contacts the update nag with the version, platform and reglevel it is running.
- The update nag then uses the versions and policies table to figure out the latest version for the platform the client is running.
- The client is also notified if it needs to pay (full price, free or discounted price) for the latest version based on the reglevel info from the client ( We allow for clients who bought the product to upgrade for free within the next 12 months and a discount price within 24 months ).
- The update nag also figures out which version the client qualifies for a free upgrade if it is not eligible to get the latest version for free.
- Further the update nag notifies release users with release versions only and beta users with both beta and release versions.
- Users also have an option on the webpage returned to the client to turn off the update nag.

## **Versions Table**

The versions table consists of one row per release of software. The software being released has certain selection criteria (such as version number, language, and mode) to determine which users it is appropriate for. It can consist of many components; the software itself, release notes, readmes, etc.

The fields are:

***file\_id*** This is a numeric id for a given file. Must be unique across the database.

***languagecode*** This is the two-character language code to identify which of our supported languages the version is for.

***product*** The name of the product. Currently only Eudora

***platform*** Where it runs. WIN or MAC

***version\_num*** The version number, in dotted decimal.

***type*** The release type; either BETA or RELEASE.

***mode*** Describes what mode the software runs in. PAY means it is a paid-for product only. FREE means it is freeware only. AD means it is adware only. ALL means it applies to all modes.

***date*** This is the release date of the software. It is actually only the nominal release date, and should be set to the first day of the month corresponding to the month code built into the software.

***size*** How big the file is.

***expires*** For expiring betas, the date they expire.

***contact\_info*** Where users with problems to report should go.

***ARL*** This is just a prefix to any fields which use URL's. Now that we're no longer using akamai, we don't need to use this.

***URL*** URL to where the software itself is.

***whatsnew*** URL to a document describing what is new in the software.

***note\_1*** Random text that goes with the software.

***note\_2*** Random text that goes with the software.

***note\_3*** Random text that goes with the software.

***note\_4*** Random text that goes with the software.

***patchURL*** URL to an updater that can be used instead of the full installer. Not used anymore

***patchURLName*** The name to put on the abovementioned updater. Not used anymore

***patchsize*** Size of the updater. Not used anymore

***patchnote\_1*** Random text that goes with the updater. Not used anymore

***patchnote\_2*** Random text that goes with the updater. Not used anymore

***manualURL*** URL to where the manual lives.

***manualURLName*** Name to put on the abovementioned manual.

***manualSize*** How big the manual is.

***manualnote\_1*** Random text to go with the manual.

***manualnote\_2*** Random text to go with the manual.

## Policies Table

The policies table in the adserver database informs the update script about how policy codes work. It makes queries to the table based on information about the version described

in the versions table and the registration information provided in the user's query.

***id*** Just an id number.

***policycode*** The policy code to which this policy entry refers.

***startmonth*** The first month at which the policycode is considered valid.

***endmonth*** The last month at which the policycode is considered valid.

***versionprefix*** The version (of the software making the request) to which this policy entry refers.

***mode*** The mode (of the software making the request) to which this policy entry refers.

***discountversionprefix*** The version (of the software making the request) to which this policy entry refers for purposes of discounts.

***discountvalue*** How much of a discount we will give to qualifying versions.

***notes*** Notes for our own use, so we can allegedly make sense of the table.

## Properties File

The jumpzone.properties file controls several parameters for the update servlet. One of those parameters, the beta prefix, may be relevant when you update the update nag, so you may also need to edit this file. Find the line:

servlet.updates.initArgs=betaprefix=

The purpose of the betaprefix is to allow us to announce new betas to people already using beta versions, but not to show those beta versions to users of released software. As new

versions of Eudora go into beta, the betaprefix should be updated to be whatever version of Eudora is currently in beta. As a version leaves beta, a higher number should be put in the betaprefix.

### **Updating the update nag**

- Entries for the various release & beta versions (both Windows and Mac) need to be added to the versions table via the versions.sql file.
- Update the database using the following command
  - /usr/local/mysql/bin/mysql -u<user name> -p<password> installers <versions.sql
- Entries for the versions and their related policy and month codes should be added to the policies.sql file
  - The start month here refers to the month code that would be eligible to get this version for free ( release month code – 12, since clients can upgrade to the latest version for free for 1 year )
  - Each release has 6 entries with different policy codes of 35-40
- Update the database using the following command
  - /usr/local/mysql/bin/mysql -u<user name> -p<password> adserver <policies.sql
- We also need to update the jumpzone.properties file in certain cases
  - The property ( servlet.updates.initArgs=betaprefix= ) has to be updated such that the betaprefix should be whatever version of Eudora is currently in beta or when we make a release the beta prefix should be a higher number than the release version.

### **Testing the update nag from a web browser**

It's possible to test the update nag from a web browser. You merely need to type in the proper update URL. That URL looks as follows:

[http://jump.eudora.com/live/updates?action=archived&product=Eudora&version=7.0.0.16&mode=paid&reglevel=170761968&la=en&platform=Windows\\_XP\\_Service\\_Pack\\_2](http://jump.eudora.com/live/updates?action=archived&product=Eudora&version=7.0.0.16&mode=paid&reglevel=170761968&la=en&platform=Windows_XP_Service_Pack_2)

The parameters in the URL mean as follows:

***reglevel*** Is computed as  $19 \times (256 \times (256 \times \text{Random} + \text{Policy}) + \text{RegMonth})$ , where Random is a random number between 0 and 255, inclusive, Policy is the policy code and RegMonth is the month registered in.

***platform*** Is the platform for which the update is requested. It will be normalized to one of the platforms listed for the versions table.

***product*** The name of the product, for matching against the versions table.

***version*** The version to be updated.

***mode*** The mode in which the version is currently running.