

# releaseManagerDaemon

The *releaseManagerDaemon* is a process that runs nearly continuously on fermilnx-v03 and forms the heart of the SCon Release Manager. The process itself is launched via standard cron every eight hours. After eight hours it shuts itself down in preparation for the new program to launch. The *releaseManagerDaemon* used to run for 24 hours but changes in the mysql server started causing all connections to fail unrecoverably after eight hours so the time was shortened. This has little impact on the running of the Release Manager.

When invoked the *releaseManagerDaemon* generates a list of all of the packages, OSes, version types, and variants available in the Release manager system by reading the relevant database tables (*package*, *versionType*, *os*, and *variant*). For each combination of these it sets up a series of processes to handle the creation of new tags (Integration versions only), tag submission, tag deletion, build submission, build cleaning, and build erasing. These processes each run on their own time scales to do their assigned task. Each is described below.

## LatestCreator

This process is set up only for the Integration version type of each package. This process runs once every 10 minutes for each of the packages.

When running, the Latest Creator looks at each of the subpackages for the specified package and looks to see if any of the tags on the subpackage have changed relative to the most recent LATEST build. If they have a new LATEST tag is created and applied to the package. If not, the process shuts down and waits to check again.

## TagSubmitter

This process is set up for each package/versionType combination and each instance runs once every 10 minutes. When it runs, the process gets a list of the tags on the package's SConstruct file and finds all that match the **versionFormat** string in the *settings* database. For each tag in this list, it checks the *buildPackage* database to see if the tag has already been added to the system. If not, an entry is added into the *buildPackage* database for the new tag.

## TagEraser

This process is set up for each package/versionType combination and each instance runs once every 10 minutes. If the **keepNumTags** setting doesn't exist for this package/versionType combination, nothing is done. If it does exist, and there are more than **keepNumTags** packages marked with **cvsTagger** = 'yes' (in the *buildPackage* table), then tags are removed from the package until the number is reduced to the value specified by **keepNumTags**.

## BuildCleaner

This process is set up for each package/versionType combination and each instance runs once every 10 minutes. If the **maxHasSource** setting doesn't exist for this package/versionType combination nothing is done. If it does exist, and there are more than **maxHasSource** distinct packages marked with **hasSource** = 'yes' (in the *build* table), then the Clean workflow step is invoked (triggering the *cleanBuild* program) for all older builds until the number of build packages is equal to the **maxHasSource** value.

## BuildEraser

This process is set up for each package/versionType combination and each instance runs once every 10 minutes. This functions identically to the BuildCleaner process but triggers off of the **maxKeep** setting (instead of **maxHasSource**) and invokes the Erase workflow step (calling the *eraseBuild* program) instead of the Clean step.

## BuildSubmitter

Unlike the other processes which are only created for each package/versionType combination, this process is set up for every package/versionType/variant/os combination as it handles the actual submission and launch of builds for each build. Like the other is is set to run once every 10 minutes.

It starts by getting a list of all build packages (*buildPackage* table) for which no build exists in the *build* table. For each of these builds it creates an entry in the *build* table and then launches the workflow step specified by the **workflowStart** setting in the *settings* table.