

build

The **build** table of the *rd_releasemgr* database holds the data on each individual build job performed by the RM and its associated **buildID**. A build is uniquely distinguished by its **buildPackageID**, **osId** and **variantId**, linking to entries in the **buildPackage**, **os**, and **variant** tables respectively. For any given **buildPackageID**, there are 1-12 builds in this table depending on the versionType and package of the build package

This table hold a lot of data on each build. The columns in this table are:

- **buildId** - The unique ID for this build. This value is used as the **subId** value for log entries in the **entry** table
- **buildPackageId** - Key linking this build to a particular build package in the **buildPackage** table
- **osId** - which os (from the **os** table) this build is for
- **variantId** - which variant (Debug/Optimized) from the **variant** table this build is for
- **workflowId** - The **workflowId** of this build. This is a key that identifies and links the build to the *rd_workflow* database tables.
- **startTime** - The time the build was initially submitted for processing. This entry is filled in by the **releaseManagerDameon** when the build is created (and possibly also from the **triggerBuild** program)
- **stopTime** - The time the build completed all processing. I believe this is filled in by the **finishBuild** program
- **checkoutStartTime** - The time the **checkoutBuild** program started running
- **checkoutStopTime** - The time the **checkoutBuild** program completed the initial CVS checkout
- **checkoutReturnCode** - The return code generated by the **checkoutBuild** program
- **compileStartTime** - The time when the **compileBuild** program started
- **compileStopTime** - The time when the **compileBuild** program completed compiling the software
- **compileReturnCode** - The return code generated by the **compileBuild** program
- **testStartTime** - The time the **testBuild** program started
- **testStopTime** - The time the **testBuild** program completed all the software unit tests.
- **buildLocation** - The physical location on disks where the software was checkout and built.
- **sconsLocation** - The path to the SCons build tool to use to build the software
- **externalsLocation** - The path to the top level of the external libraries to be used in the build.
- **userReleaseLocation** - The path the location of the downloadable user release distribution file (tar or zip depending on OS)
- **develReleaseLocation** -The path the location of the downloadable developer release distribution file (tar or zip depending on OS)
- **sourceReleaseLocation** -The path the location of the downloadable source release distribution file (tar or zip depending on OS)
- **hasSource** - A flag indicating whether or not the source code is still on disk in the **buildLocation**
- **visible** - A flag indicating whether or not the build should be visible in the build viewing tools (RMViewer, webpages) if other viewing criteria are met.