Mt Wilson

Automated Build

# Background

Mt Wilson has a lot of components in multiple programming languages and target output formats. The build must be automated and simple for developers to understand and use.

# Architecture

Maven is used to build individual projects and copy their build products to the Maven repository, where they can be referenced by other projects -- possibly in other Git repositories.

Ant is used to provide an easier interface to start a build, specify what should be the output, and initialize Maven. There is a build.xml file at the root of each Git repository which is capable of directing a complete build. The command “ant -projecthelp” shows the available targets. The default target should be a clean build with all final products included.

Automated builds are centralized on a build server which acts as a Team City “agent”. Builds are numbered sequentially, and each active branch in the Git repository is built separately so there can be multiple builds in each project. Builds are organized by Git repository and branch.

There is a build.targets file at the root of each Git repository which lists the expected outputs of a successful build using paths relative to the location of the build.targets file at root of the Git repository. The build.targets file itself may be generated by the build process in order to dynamically set the project version number in the expected outputs.

There should be a “packages” folder at the root of each Git repository which contains projects that compile the distributable packages which should be the final build products from that repository. This allows developers to conveniently skip building the final packages when it’s unnecessary, and also makes it convenient to create variants that are clearly separated from the main code base.

## Maven

Each directory built by maven has a pom.xml file. This file acts as Maven’s directory listing, and specifies which subdirectories have Maven projects that should be built.

There is a root pom.xml file called mtwilson-maven-root. This pom.xml file has general project metadata, dependency version management that is needed throughout the project, and minimal build instructions that are common to every project.

There is also a set of pom.xml files whose names start with mtwilson-maven and they are used to provide more specialized build instructions for projects. The pom.xml named mtwilson-maven-java has instructions for building all Java projects. The pom.xml named mtwilson-maven-package-zip has instructions for building a zip file containing all project dependencies (transitive) and static resources. The pom.xml named maven-package-makeself has instructions for building a self-extracting linux archive containing only specified dependencies (non-transitive). These mtwilson-maven projects do not produce any useful artifacts on their own apart from preventing duplication of the same build instructions throughout dozens of similar projects.

# Operation

To build the final products in a Git repository, check out the relevant branch and run “ant”. This should produce all the files listed in “build.targets”.

In general, “ant clean” should translate to “mvn clean”, and “ant build” should translate to “mvn install”. Running “ant packages” should build all the distributable packages but is not required to produce all the files listed in “build.targets”.