Makefile Conventions

1. Introduction

This is a brief description of the current makefile conventions. In its final form, this material should be integrated into the SOP document.

2. Location of Makefiles

In the UNIX environment, most project (sub)directories contain Makefiles. As necessary, Makefiles at a higher level descend into subdirectories that eventually contain actual target files to be made. For example, the Makefile in the implementation directory descends into platform-specific directories, as appropriate for a particular project.

3. Contents of the Configuration Subdirectory

To be fully consistent with Inferno conventions, it would be appropriate to store the Makefiles not in project subdirectories directly, but rather in a *configuration* subdirectory. However, this would make path definitions within the Makefile unnecessarily confusing. Hence, as a pragmatic matter, we store a Makefile in the directory to which it applies. This is analogous to the existence of the CVS UNIX subdirectories, that are not explicitly part of the Inferno artifact hierarchy, but which exist as required for tool support.

If a Makefile employs any auxiliary script files, these *are* stored in the configuration subdirectory, immediately below the directory in which the Makefile is stored. In this way, only the Makefile itself violates the Inferno subdirectory conventions, with any other configuration-related files stored in the `configuration` subdirectory.

4. Making in Release Directories

As noted in the CVS conventions document, all purely generated files are declared to be ignored by CVS. Hence, when a work directory is checked in, it does not include any generated files. Therefore, when a release directory is updated, part of the update process must be a full make of the project. This convention requires that all purely generated files must be accounted for in the Makefile hierarchy. That is, a generated file cannot only be made "by hand", outside of the context of a Makefile. Certainly hand building of artifacts is fine during the course of normal work. However, the point here is that every purely generated file must be accounted for in a Makefile. In this way, all purely generated files that are ignored by CVS will be made by the invocation of the top-level project Makefile.