DylanWorks Listeners

Scott McKay & Roger Jarrett

1. Introduction

The paper describes the functionality of “listeners” in DylanWorks model.

2. Basic Functionality

A traditional “listener” is a tool that implements a “read/eval/print” loop, that is, it reads a language form from a user, evaluates the form, and then prints the result.

In DylanWorks, the model is somewhat richer. In addition to reading language forms, listeners also function as general purpose “command processors”. Much of the functionality found in other tools in DylanWorks will be accessible directly from a listener, although the interface to such functionality in a listener will be different. For example, in a listener:

- Class browsing functionality will have a command-line interface.
- Generic function and method browsing functionality will have a command-line interface.
- Cross-reference browsing functionality will have a command-line interface.
- Compilation and loading of files and projects will have a command-line interface.
- There will be command-line interfaces to many “utility” functions, such as getting directory listings, renaming files, and so on.

The main interaction window of a listener will also be based on a presentation-recording stream, so that all output done in a listener is “active”. The combination of “active” output and imported functionality will cause listeners to behave as an “all-purpose” tool.