# PVS Plugin Design Notes

### Syntax Highlighting

The java package containing the files is “com.sri.csl.pvs.plugin.editor”. The main class is PVSScanner.java where all the rules for the syntax highlighting are added. There are currently five categories of highlighting:

* Keywords such as CONJECTURE, FORMULA, and EXISTS
* Constants including Strings and Numbers
* Operators such as =>, ==, and &&
* Comments starting with %
* Default which is anything else

PVSWhitespaceDetector defines white spaces. PVSWordDetector defines what an acceptable word for PVS is.

## Editor

PVSEditor.java defines the pvs editor. The extension “.pvs’ is specified in plugin.xml and under the extension that defines “PVS Editor”.

The listener that catches events such as EditorOpen, EditorActivated, etc is PVSEditorActivationListener.

## Preferences

The packaged containing the classes related to preference page is “com.sri.csl.pvs.plugin.preferences”. The class PVSPreferencePage contains a method called createFieldEditors, which creates all the desired fields in the preference page.

## Console

PVSConsole defines a new console in Eclipse with both input and output capability.

IOConsoleKeyboardReader is a java class that monitors the pvs console for any line typed by the user and passes it to the running pvs.

## Theories View

PVSTheoriesView is a view that shows the theories and the declarations in a tree form. Its content changes whenever the user makes a pvs file as active. This tree model should be created first (using TreeNode.java). The new model can then be passed to the viewer via PVSTheoriesView.getInstance().setInput(newModel);

## Running PVS as an External Process

## PVS Menu

## PVS Toolbar