Rsldoc Design Overview

The `rsldoc` documentation generator is written in C++. The `.h` files contain further design documentation. All of the `.h` files, as well as the matching `.C` files, are in the sibling implementation source directory,

The top-level class is RSLDoc, defined in The design comment in this class describes the overall organization of the documentation generator, both current and planned.

Class `RSLToDict` is the major work-doing class. It performs the main task of generating a data dictionary from the rsl source text. It's design documentation is in

`RSLToDict` does its work by calling the RSL translator to generate a symbol table. The table contains the definitions of all identifiers in an RSL spec, most particularly the modules, objects, and operations. The documentation generation functions traverse the symbol table, extract the appropriate information, and generate the HTML files that define the user-viewable data dictionary.

The other major class in the generation process is `TransInterface`. It is a wrapper around the RSL translator, used by the functions in `RSLToDict`. It's defined in

In a number of places in the design documentation, there are references to a tool called "*the browser*". These are historical references to an earlier version of the `rsldoc` tool that displayed RSL documentation in an custom X-Windows browser, rather than generating HTML that can be viewed in a standard web browser. For the purposes of design, the `rsldoc` generator and "*the browser*" are the same thing.

The design documentation in the `.h` files is in a form very similar to `javadoc`-style comments. There is a class comment at the top of the `.h` file. Then for each function (aka, method), there is API documentation, describing what the function does, its inputs, and outputs. Syntactically, the function comments come *after* the function signature, rather than before it as with `javadoc` comments.