Enhancements to the Rsldoc Browser

Revised rsldoc layout:

```
-----------------------------------------------
| Modules:     |  -- presentation options --  |
|   ...        |                              |
|              |    Main Presentation Area    |
|              |                              |
|-------------|                              |
| Objects:     |                              |
|   ...        |                              |
|              |                              |
|              |                              |
|-------------|                              |
| Operations:  |------------------------------|
|   ...        |      Entity Details Area     |
|              |                              |
|              |                              |
-----------------------------------------------|
```

A la javadoc, the main presentation area flips between a variety of info, as does the details area, based on the presentation options, which are the following:

```
Overview   Module   Hierarchy   Dictionary   UML   DFD   Source   Index   Help
                 Alphabetic Order   As-Declared Order
          Case-Sensitive Ordering  Case-Insensitive Ordering
          Prev Next   Details No Details   Frames No Frames
```

The **`Overview`** display shows a two-column table of modules, very similar to the javadoc **`Overview`**. The first column contains clickable module names. The second column contains the first sentence of the module description.

The **`Module`** view shows two two-column tables, similar to the **`Package`** view in javadoc. The two tables show objects followed by operations. The first column in each table is the clickable name, the second column is the first sentence of the description.

The **`Hierarchy`** view is the same as what is shown in the lower right frame in the current rsldoc. Clicking on an object in the hierarchy view swaps in the dictionary view and scrolls it so that the selected object is at the top This is the same behavior as the current rsldoc, except in the new version the hierarchy is replaced with the dictionary in the main frame, rather than having the hierarchy remain visible in the lower frame. The hierarchy should have added functionality that allows levels to be expanded and collapsed, a la Windows Explorer.

The **`UML`** and **`DFD`** views show the diagram for the currently selected module. Clicking on an object or operation name in a list frame (on the left of the display), pans the diagram so that the selected object or operation is centered in the presentation area. Clicking on an object or operation in a diagram swaps in the dictionary and scrolls to the selected item.

The **`Source`** view shows the rsl code for the currently selected module. Clicking on an object or operation in a list frame scrolls to its source definition. All object and operation names in the source view are links to the dictionary, as they are in the **`Hierarchy`** and diagram views.

The **Index** and **Help** views are a la javadoc.

Unlike the current version of rsldoc, there should be an "All Modules" selection in the Modules frame, as there is an "All Classes" selection in the javadoc classes frame. This means that the Modules frame acts basically like the javadoc packages frame in that selecting "All Modules" means that the Objs and Ops frames list all defined entities, whereas selecting an individual module from the list in the Modules frame means the Objs and Ops frames list only the entities in the selected module. The same applies to the other display windows as well, i.e., the main presentation and details areas display only the entities in the selected module.

Whenever the user selects an obj or op in the main frame, its details appear in the details frame. The details area appears and disappears depending on the current option setting. Specifically, the details areas is visible for all presentations except **Overview**, **Index**, and **Help**. The convention is that whenever the main frame contains selectable modules, objects, or operations, the details frame is visible and contains the detail information for the current selected item.

For modules, the details are:

```
Defined in: m.rsl
Number of Objects: n
Number of Operations: n
Number of Auxiliary Functions: n
Number of Attribute Definitions: n
Number of Axioms: n
Number of Theorems: n
```

where m.rsl is a link that opens the module file in a new browser window.

For objects, details are:

```
Defined in Module(s): M (m.rsl), ...
Component of Object(s): O, ...
Inherited from: O, ...
Operations: Op, ...
Equations: e ...
Input by: Op, ...
Output by: Op, ...
Transitive Number of Components: n
Where: ...
Defined Relations: ...
Defined Attributes: a, ...
```

where all the things that can/should be links are.

For operations, details are:

```
Defined in Module(s): M (m.rsl), ...
Precondition: p
Postcondition: p
Components: ... (this line absent if none)
Dataflow: ... (ditto)
Component of Operation(s): Op, ... (ditto)
Inheritees: O, ...
Transitive Number of Components: n
?Where: ...?
Defined Relations: ...
```

```
Defined Attributes: a, ...
```

where all the things that can/should be links are.

The way to see both a data dict and a UML diagram side by side is to open up two browser windows with the dict view selected in one and the UML view shown in the other.

18mar03

1. Horz scrollbars are fine.
2. Use module name and entity name highlighting instead of the Viewing and Item boxes.
3. The `Module`, `UML`, `DFD`, and `Source` links are disabled when ‘All Modules’ is selected.
4. Here’s what’s displayed for the UML and DFD links:
   a. There’s a one-to-one correspondence between .rsl files and UML and DFD files.
   b. The UML file for X.rsl is named X.uml.{jpg,gif,png,...}.
   c. The DFD file for X.rsl is named X.dfd.{jpg,gif,png,...}.
   d. Both UML and DFD images are drawn by hand using some external tool.
   e. The UML and DFD links are disabled when ‘All Modules’ is selected or when any module that has no .uml.* or .dfd.* (respectively) file defined.
   f. If there’s more than X.{uml,dfd}.* defined, then it will use them in the following order: .jpg, .gif, .png, ...
5. Collect all non-module defs into a single file named "_Non-Module-Definitions" and display this as a normal module in the browsers.