Rich Tree Operations

Rich Trees are call-trees or caller-trees where each function-node contains additional information for easy browsing and understanding of project code.

With the additional information, you can see:

a) in what functions, a particular global variable is used

b) in what functions, a particular struct or class is used

c) what members of a struct or class are referenced or modified by those functions

d) what parameters are needed by each function

e) in what files, are the functions defined

f) the comment description of each function alongside each node in the tree

Examples:

Suppose you just created the call-tree of main( ) and you wish to see

1) which functions reference a particular global variable:

- Simply search for the global variable in the call-tree
- Crystal will display the global variable next to each node that uses the global variable.

2) which members of the struct or class are used by the functions.

- Crystal will display the structure/class members next to each node that references those members

3) Crystal Flow/REVS picks up a function's description from its comments

- You can get a quick overview of a call-sequence in the tree by viewing the functions' description alongside the tree-nodes.

Rich tree is integrated with DataFlow:

Having identified the functions that use a particular global, struct, macro etc.:

- Create the DataFlow and view how that global, struct or macro is used in those functions

Create a Call-tree or Caller-tree as usual	– Place the cursor within a function; click the Call-tree icon – Place the cursor on a function name; click the caller-tree icon
Display the "Search in Tree" menubox	– Click the "Search in Tree" icon in the Tree Window toolbar
♦ The "Object Search" card is used to:	– Find - in what functions, a particular data object, struct name, class name, type name or #define is used.
• Specify the data object or struct name etc.:	– In the "Object Search" card: ▪ Select the Type of Object ▪ Use the "Select From" drop-down list and/or the keyboard to specify the name of the object, struct name etc. (when function's filenames are being displayed) ▪ When you specify a struct name or class name or an instance of struct or class: You have the option of searching for a member If you wish to search for a member: 1. Use the "Members" drop-down list to enter the "**.**" or "**->**" operator 2. Use the "Select From" drop-down list and/or the keyboard to specify the member ▪ "Object Name" contains the string resulting from the steps above. The contents of "Object Name" will be searched for. ▪ Alternatively, you can use the drop-down history list of "Object Name" ▪ You can click the <Reset> button and start over.
• Find -in what functions, the data object is used:	– After specifying the data object or struct name etc.: a) Click the <Search> button. Crystal will search to find the next function where the specified data object is used. If the function is not visible in the tree, Crystal will expand the appropriate branch in the tree so that the

	function becomes visible. b) Or Click the <Search All> button. Crystal will search to find all the functions where the specified data object is used. If the functions are not visible in the tree, Crystal will expand the appropriate branches in the tree so that the functions become visible. Further, to make it easier to view the results, it will
	automatically perform "Show All Paths" to the found functions, i.e. it will hide all nodes that are not in the path to the found functions. To view the whole tree again, right-click anywhere in the Tree Window, then click on <Show Whole Tree> in the pop-up menu. Note: Crystal expands only those branches that contain the "found" functions. This will help you view only those paths that are of interest. If you wish to view the whole tree: Expand the tree fully before the search: Click on the root node to select it. Then right-click anywhere in the Tree Window and click <Expand Full> in the pop-up menu.
♦ Use the "Advanced Search" card to:	– Search for: ▪ the next occurrence of a function in the tree Select the "Function Name" checkbox, Use the history drop-down list or the drop-down list of function names and/or the keyboard to enter the function name ▪ the next occurrence of a filename in the tree (when function's filenames are being displayed) Select the "Function Name" checkbox … ▪ Use the "history list" to search for a particular data object, struct name, class name, type name or #define ▪ string search in anything that is being displayed in the tree-nodes. i.e. the function-name, filename, parameter list etc. You can select more than one checkbox. Click the <Search> button to find the next node containing a match for each of the selected item.

♦ The "Enable Display" card:	
– Show with each node	– Enable/disable what is displayed alongside each node
– Highlight other occurrences of	– Select the criterion by which other occurrences of the current node are highlighted You can select more than one item. Crystal highlights other nodes in the tree that match the current node in the items selected

Other Operations:

View the DataFlow of the data object:	After searching for functions that use the data object (as described above): – Click to select a function-node that has the data-object displayed alongside it – Right-click anywhere in the Tree Window, then click <DataFlow for the Current Object> in the pop-up menu. Crystal will display the DataFlow of the data-object in that function. You can expand the function-calls in the DataFlow to dig deeper.
– Go to Function Call on Single Click	Use the "Project" or "Tools" pull-down menu to go to the Options menu In the Options menu, go to Options -> Customize -> Tree Select " Go to Function Call on Single Click…", Click <OK>