

# Crystal C/C++


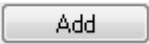
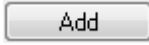




## Quick Reference Guide

(This page is intentionally blank)

## Index

1. CREATE A PROJECT .....	4
2. DUMMY-HEADER FILE & LANGUAGE EXTENSIONS .....	5
3. FLOWCHART .....	6
4. TREES – CALL, CALLER, FILE, BASE, DERIVED, PROJECT, RELATIONSHIP .....	10
5. TREE OPERATIONS .....	12
6. BROWSE WINDOW .....	16
7. NAVIGATION .....	18
8. RICH TREES .....	22
9. DATA FLOW .....	27
10. CALL FLOW .....	29
11. DATA-DEPENDENCY TREES .....	31
12. GENERATING HTML DOCUMENTATION .....	34
13. STATIC CHECKING .....	37
14. COMMENT TRAIL .....	39
15. STATE TRANSITION DIAGRAMS .....	40
16. EDIT OPERATIONS .....	42
17. FORMATTING FOR IMPROVED READABILITY .....	44
18. BATCH OPERATIONS .....	45
19. MISCELLANEOUS OPERATIONS .....	46
20. TROUBLE SHOOTING .....	47

## 1. Create a Project

	By creating a project, you can browse across files.	
1.	Create a project	<p>Click File→ New Project</p> <p><b>Location:</b> The folder where you wish to create the project file</p> <p><b>Project Name:</b> Name of the project file you wish to create. The project file will have the extension .sur or .pur</p>
2.	If source files or header files have extensions other than *.c, *.cpp, *.h	
3.	Add source files and header files to the project	<p>Click  &lt; Add Files &gt; in Project Management Menu.</p> <p>Select the radio button "All Source Files" and Click </p> <p>Or Select files from the displayed list and click </p>
4.	Set the Include-Path-Sequence	<p>Click  &lt; Include Path &gt; in the Project Management Menu.</p> <p>It is not enough to simply add the header files to the project...</p> <ul style="list-style-type: none"> <li>– Select the “Include subfolders” box if header files are #included from the sub-folders too.</li> </ul>
5.	Parse All Files	Click  < Parse All > in the Project Management Menu.
6.	After parsing, check for Missing header files.	<p>Click  &lt; Missing Includes &gt; in the Project Management Menu</p> <p>If header files are missing, some type names may remain undefined and result in syntax errors.</p> <p>Syntax errors are displayed in pink color.</p> <p>If a syntax error involves a function-header (start of a function), then the function is not recognized and hence its flowchart is not available.</p>
7.	If any header files are missing:	Add the necessary folders to the Include-Path-Sequence so that Crystal Flow can find the header files
8.	After adding folders to the Include-Path-Sequence:	Reparse All Files by clicking  < Reparse All > in the Project Management Menu.

## 2. Dummy-Header File & Language Extensions

1. If your code contains non-ANSI C keywords:

Use the dummy-header-file or the Language extensions to handle non-ANSI C keywords:

Language extensions can be supported by means of either the dummy-header file or the language extensions card of the Options menu:

Let us start with an example:

```
__user TBool find_ele( int *arr_p, int amax )
{
...
}
```

a) Since ANSI C does not recognize `__user` and `TBool`, you can add a declaration or `#define` for `__user` and `TBool` in the Project Dummy Header file. Crystal REVS automatically `#includes` the Project Dummy Header file at the beginning of each file in the project.

You can open the Project Dummy Header file from the "Project" pull-down menu or the "Prj Files" pane of the Browse Window.


The Project Dummy Header file is like any other header file, you can edit it and save it. You can `#include` other files in it.

```
#ifndef dummy_header_is_included_once
#define dummy_header_is_included_once 1

# define __user
typedef int TBool;

#endif
```












After you modify the dummy-header file, save it and Reparse All files.







b) Or you can use the Language Extensions card of the Options Menu. You can add a new scheme. Enter `TBool` in the "Types" and click . Similarly add `__user` to the "Others" column.







(Adding to the "Others" column has the same effect as `#define __user` as shown above.)

Both Project Dummy Header file and Options settings are saved on a per-project-basis.



### 3. Flowchart

1.	Create a Flowchart	Place the cursor anywhere in a function. Click < <a href="#">Create Flowchart</a> >  in the toolbar or press 
2.	View Flowchart and Code side-by-side  <a href="#">Highlight code in File window</a>  <a href="#">Scroll highlighted code in File window</a>	Click on the  button below the window to display the code and flowchart side by side.  Click in the <b>right</b> -half of a flowchart symbol to highlight the corresponding code in the File window.  While the corresponding code is highlighted, click the scroll icons  or  in the flowchart toolbar
3.	View Type Information	a. Click in the left half of a flowchart symbol. b. Hover the mouse pointer over an object name in flowchart symbol.
4.	View the conditions that need to be true	To view the conditions that need to be true in order to reach the current flowchart symbol:  - Select < <a href="#">Show/Hide Outer Window</a> >  in the toolbar at the top of the condensed-flowchart.  In the sequence of conditions that are displayed below the condensed flowchart  - Click on any condition in the sequence to go to the corresponding symbol.
5.	Flowchart Toolbar	Roll the mouse pointer over toolbar buttons to gain familiarity
6.	High-level Symbols  <a href="#">View Code Coverage</a>  <a href="#">Expand a high-level symbol</a>  <a href="#">Collapse symbols at the current level</a>	A high-level symbol is indicated <b>by a thick purple border</b> .  <a href="#">Click in the left part of a high level symbol to view the code covered by it.</a>  <a href="#">Double click on a high-level symbol to expand it.</a>  <a href="#">Double-click in the right-half of a low-level symbol to collapse all symbols at that level</a>
7.	Level-n-flowcharts	Click  ,  ,  or  so that the condensed view shows the "right" amount of detail; neither oversimplified nor too crowded.  Click the < <a href="#">Create an optimal Flowchart</a> >  to view the default level.
8.	Divide and Conquer	For long functions, get the whole view of the function by seeing it's top level flowchart. Then view the flowcharts of it's major parts such as large switch-statements, loops, etc





9.	<p>Create a sub-flowchart</p> <p>Create a switch-flowchart (Similarly create the flowchart of a while-loop, if-else, for-loop etc.)</p> <p>Create a deeper sub-flowchart</p> <p>Go back to parent flowchart</p> <p>Go back to sub-flowchart</p>	<p>Click , the 2<sup>nd</sup> icon from the top in the Flowchart window toolbar</p> <ul style="list-style-type: none"> <li>- when a "switch" symbol in the flowchart is selected, the  icon is called "Create Current Switch Flowchart"... similarly, for a "while" symbol, a "for" symbol etc.</li> </ul> <p>a. Place the cursor on a switch statement in File window; click Flowchart → Create Code Flowchart of → <b>switch</b></p> <p>b. Click on a switch symbol in the flowchart; click &lt; Create Current Switch Flowchart &gt;  i.e. the 2<sup>nd</sup> icon in the toolbar</p> <p>In current sub-flowchart, select the "right" level of detail with <b>L<sup>1</sup></b>, <b>L<sup>2</sup></b>, then create a sub-flowchart as needed</p> <p>Click  in the toolbar of the Flowchart window</p> <p>Click  in the toolbar of the Flowchart window</p>
10.	Go to the flowchart of a Called function	<p>Right-click in a symbol that contains a function-call.</p> <p>Click &lt; Create Flowchart for function call &gt; then click the 'function name'.</p> <p>Use &lt; Change mode &gt;  near the bottom of the toolbar to go back and forth between the caller-flowchart and called-flowchart</p>
11.	<p>Expand function-calls in a flowchart</p> <p>Create an expandable Flowchart</p> <p>Expand a function-call</p> <p>Collapse an expansion</p>	<p>First, you need to create an expandable flowchart as described below.</p> <p>Place the cursor anywhere in the function.</p> <p>Click the pull-down menu CallFlow→Expand Calls in a Full Function Flowchart</p> <p>The presence of a function-call is indicated by the  above the symbol.</p> <ul style="list-style-type: none"> <li>- Double-click on a symbol that contains a function call.</li> </ul> <p>The expansion is inserted just before the function call.</p> <p>Double-click on the start symbol of an expansion</p>
12.	Transparent Flowchart Window	<p>When the flowchart window overlaps the File window, you can see the source lines covered by the flowchart:</p> <ul style="list-style-type: none"> <li>- Click  in the flowchart window toolbar to make the flowchart window transparent.</li> </ul>



13.	Navigating in Flowchart	
	<p>Arrow-keys and the Home key</p> <p>Hand icon to drag and move</p>	<ul style="list-style-type: none"> <li>- Press the arrow keys to walk through the flowchart. Press the Home key to go to the start of flowchart</li> <li>- Click  to select it. Now you can click-and-drag to move in the flowchart.</li> </ul>
14.	<p>Highlight one or more connections</p> <p>Highlight a Flowchart connection</p> <p>All connections that arrive at a statement</p> <p>All paths that leave a high-level symbol</p> <p>Highlight another connection</p>	<p>Click on a connection to highlight it.</p> <p>Click on the input of a flowchart symbol. All paths that go into the input are highlighted.</p> <p>Click on the output of a symbol. All paths going out of that output are highlighted.</p> <p>To highlight another connection while retaining the current highlighting, press  and then click on a connection to highlight it.</p>
15.	Visual distinction in a monotonous flowchart	Please view the Help contents. (Flowcharts->Basic Operations->How to add...)
16.	Text search in a flowchart	<ul style="list-style-type: none"> <li>- When in a flowchart, click  in the main toolbar.</li> <li>- Enter the "text" to be searched.</li> <li>- Click &lt; Mark All &gt; - the symbols with the matching text are highlighter.</li> <li>- Use  and  to view the symbols containing the matching strings</li> </ul>
17.	<p>Customize</p> <p>Labelling of True and False branches</p> <p>Flowchart Settings</p> <p>Flowchart Appearance</p> <p>Customize Flowchart window</p>	<p>You can customize the text displayed on the YES and NO paths of an "if statement."</p> <ul style="list-style-type: none"> <li>- Click Flowchart →Customize to view the options menu</li> </ul> <p>The flowchart is controlled and customized by several flowchart-settings.</p> <ul style="list-style-type: none"> <li>- Click &lt; Flowchart Settings &gt;  in the flowchart toolbar to modify settings.</li> </ul> <p>Click Flowchart →Appearance to customize the look and appearance of the flowchart</p> <p>To customize the fonts, font-size, toolbar position, connection and symbol line thickness, etc:</p> <ul style="list-style-type: none"> <li>- Click Flowchart →Customize to view the options menu</li> </ul>







	Customize shapes for system-calls	<p>Symbol shapes for function calls can be customized. To customize shapes:</p> <ul style="list-style-type: none"> <li>- Click Flowchart →Customize Shapes to view the Associate Function Names with Shapes dialog.</li> </ul>
18.	<p>Types of flowcharts</p> <p>Code Flowchart</p> <p>Comment Flowchart</p> <p>Code + Comment Flowchart</p>	<ul style="list-style-type: none"> <li>- Click &lt; Change mode &gt;  in the flowchart toolbar and then select &lt; Switch to Code Flowchart &gt;</li> </ul> <p>Place the cursor within the function</p> <ul style="list-style-type: none"> <li>- Click the pull-down menu Flowchart → Create Comment Flowchart of → current function</li> <li>- Click &lt; Change mode &gt;  and then select &lt; Switch to Code + Comment Flowchart &gt;</li> </ul>
19.	Special Comments	<p>In a comment flowchart, you can replace a loop, a switch or a few statements with an overall comment. Please view the Help contents. (Flowcharts-&gt;Special Comments...)</p>
20.	Print a flowchart	
	<p>Print preview</p> <p>Sizing for print</p>	<p>Right-click in the flowchart window; Click &lt; Print Preview &gt; in the pop-up menu.</p> <p>In the Print preview window, click the "Tile Pages" button to view multiple pages or a single page.</p> <ol style="list-style-type: none"> <li>Click "Zoom In" or "Zoom Out" in the flowchart toolbar then view the Print preview.</li> <li>You can print the whole flowchart on a single page: <ul style="list-style-type: none"> <li>- Right-click in the flowchart window; Click Print; Select "Flowchart in Single Page"</li> <li>- Use the pull-down menu File-&gt;Page Setup-&gt;Flowchart</li> </ul> </li> </ol>
21.	<p>Export Flowchart</p> <p>Export flowchart - .jpg, .bmp, .html</p> <p>Export flowchart to Visio</p>	<p>With the flowchart window active, use the pull-down menu Flowchart→Export Flowchart Image</p> <p>Similarly, you can click-and-drag to select a part of the flowchart and export it as a bitmap.</p> <p>Right-click in the flowchart window; then click "Visio Export"</p> <p>With the flowchart window active, use the pull-down menu Flowchart→ Visio Export</p>

















## 4. Trees – Call, Caller, File, Base, Derived, Project, Relationship














1.	Call Tree	Place the cursor within a function. Click  in the toolbar
2.	Caller Tree	Place the cursor on a function name. Click  in the toolbar
3.	File Call Tree	The File Call tree shows the relationship between all functions in the current file. It helps you to get started with understanding and reviewing the file. – Click Tree→Create File Call Tree
4.	File Tree	The file-tree shows files that are #included in the current file. – Click  in the main toolbar
5.	Data Dependency Tree	Place the cursor on any data-object. – Click <Create Data Dependency Tree>  icon in the toolbar.
6.	Base Class Tree	A base class tree is a tree of the current class and it's base classes. – Place the cursor on a class name. – Right click, then click Base Classes.
7.	Derived Class Tree	A derived class tree is a tree of the current class and it's derived classes. – Place the cursor on a class name. – Right click, then, click Derived Classes
8.	Include By Tree	The Include by Tree shows files that include the current file. – Place the cursor anywhere in the current file. – Click the Tree→Create Include By Tree
9.	Project Call Tree	A Project Call Tree is the Call Tree of all root functions of a project. – Make sure that all files in the Project have been parsed (in Project Management menu). – Click Tree→Project Call Tree
10.	Project File Tree	A Project File Tree is a file tree of all project files. The files not included by any file are at the first level. – Make sure that all files in the Project have been parsed (in Project Management menu). – Click Tree→Create Project File Tree

11.	Project Class Tree	<p>Make sure that all files in the Project have been parsed (in Project Management menu).</p> <ul style="list-style-type: none"> <li>– Click Tree→Create Project Classes Or press  + </li> </ul>
12.	File Relationship Tree of the current file	<p>A File Relationship Tree shows the files from which the current file uses Globals or Functions.</p> <ul style="list-style-type: none"> <li>– Place the cursor anywhere in the current file.</li> <li>– Click the Tree→Create File Relationship Tree</li> </ul>
13.	Class Relationship Tree of the current class	<p>A Class Relationship Tree displays the classes which use the Data Members or Function members of the current class. Such classes should not be in the hierarchy of the current class.</p> <ul style="list-style-type: none"> <li>– Place the cursor on a class name.</li> <li>– Click Tree→Class Relationship Tree</li> </ul>

## 5. Tree Operations

1.	Order tree-nodes by name	Click < Tree Settings >  in the tree toolbar. In the popup, click Call Tree nodes sorting→By Name Alternatively, you can use the pull-down menu Tree→Settings...
2.	Order tree nodes by call-occurrence	Click  in the tree toolbar. In the popup, click Call Tree nodes sorting→By Occurrence→Show All Occurrences  To order by the <b>first call</b> of a function and hide its other calls, click <Show First Occurrence> in the above.
3.	Show library functions, undefined functions etc.	Click  . In the popup, turn the desired item On or Off.
4.	Expand the whole tree	To <b>fully expand</b> the sub-tree under the current node, Right click anywhere in the Tree window; Click < Expand Full >
	Expand once	To keep the tree compact, tree nodes which have been expanded once are not expanded again.  - Click  in the tree toolbar. Turn < Expand Once > ON.
5.	Node Prefixes	<ul style="list-style-type: none"> <li>- <b>P</b> - indicates that the node is a pointer to a function.</li> <li>- <b>L</b> - indicates that it is a library function/library file.</li> <li>- <b>A</b> - indicates that this function was assigned to the pointer-to-function or used to initialize a pointer to function in a declaration</li> <li>- <b>*</b> - indicates that the function is recursive</li> <li>- <b>X</b> - indicates that the function is declared but not defined.</li> <li>- <b>?</b> - indicates that the function is neither declared nor defined</li> </ul>
6.	Go to function-call	A single click on a node in the Call-Tree: <ul style="list-style-type: none"> <li>- moves the cursor to the first call of the function after the current cursor position.</li> <li>- Subsequent clicks will move the cursor to the next call of the function in the current function</li> </ul>
7.	Go to function-definition	In a Call-tree or Caller Tree, double click on a node to go to the definition of that function.
8.	Type information of current node	To view the prototype (or the full declaration) of the function in the <b>current</b> node, Right click anywhere in the tree window.; Click < Type Information > in the popup menu.

9.	Search in tree	
	Using the Search box in the search toolbar	<p>With the Tree window active, use the drop-down list in the Search Panel in the main tool bar. It will show all the functions names in the tree.</p> <ul style="list-style-type: none"> <li>– Enter a first few characters of the desired function.</li> <li>– When the desired function is visible in the list, double click and enter it in the search panel.</li> <li>– Press the enter key.</li> </ul>
	Using the Search icons in the main toolbar	<p>Click  to go to the next occurrence of the <b>current</b> node in the tree.</p> <p>Click  to go to the previous occurrence of the <b>current</b> node in the tree.</p>
10.	View call-sequence	<p>a) Call-path from the root to the current node.</p> <p>b) Call-paths from root to all occurrences of the current node.</p>
	Call-sequence to current node	Right-click, then click < Show Path >.
	View all call-sequences	Right-click, then click < Show All Paths >.
	Show whole tree (call-sequence nodes are highlighted)	<p>After viewing a call-sequence,</p> <ul style="list-style-type: none"> <li>– To view the whole tree again, right-click, then click &lt;Show Whole Tree&gt;</li> <li>– To turn Off highlighting of the sequence-nodes, right-click, then click &lt;Unmark Path&gt;</li> </ul>
11.	Navigating in the tree	
	Use the Home key, Arrow keys	<p>Press  to view the root of a fairly long tree</p> <p>Press  to move downwards at the same level or to the child-node</p> <p>Press  to move upwards at the same level or to the parent node</p> <p>Press  to expand the sub-tree of the current node and move to it's first child.</p> <p>Press  to collapse the current sub-tree and go to the parent of the current node.</p> <p>Also try the  + ,  +  and the  +  key combinations.</p>
	Go to parent node	Click  in the toolbar or press the  +  key combination.




	Go back to previous / Go to next	<ul style="list-style-type: none"> <li>- Click  in the tree. Alternately, you can press  +  key combinations. When you click this icon repeatedly, you will reach the first node that was visited.</li> <li>- Click  in the tree. Alternately, you can press  +  key combinations. When you click this icon repeatedly, you will reach the last node visited.</li> </ul>
12.	Mark / hide	
	Marking and unmarking nodes	<p>To mark a node: Select the node by clicking it; Click  in the toolbar</p> <p>To unmark a node: Select the node by clicking it; Right click, then click &lt; Unmark &gt;</p>
	Go to a marked node up/down	Click  or  to go upward or downward to a marked node.
	Hiding and unhiding nodes	<p>To hide an insignificant node,</p> <ul style="list-style-type: none"> <li>- Select the node you wish to hide. Click on  in the toolbar</li> </ul> <p>To unhide a hidden node,</p> <ul style="list-style-type: none"> <li>- Click  on the toolbar and select the node you wish to unhide Click on  in the toolbar</li> </ul>
13.	Tree Settings	Click on  in the toolbar to change the settings. For details, refer to Help.
14.	Tree Appearance	Click Tree→Appearance to customize the look and appearance of the flowchart
15.	Customize the Tree Window	To customize the fonts, font-size, toolbar position, etc, Click Tree→Customize.
16.	Export Tree	
	Export as .jpg, .bmp or .html	<p>To Export a part of the tree as a Bitmap:</p> <ul style="list-style-type: none"> <li>- Select the part of the flowchart to be exported by drag and select.</li> <li>- In the flowchart window, click Tree→ Export Tree Image →Selected</li> </ul> <p>To Export a tree as a Bitmap:</p> <ul style="list-style-type: none"> <li>- In the flowchart window, click Tree→Export Tree Image→Whole.</li> </ul>

	Export as a .txt file	<p>To Export a tree as a text file:</p> <ul style="list-style-type: none"> <li>- In the tree window, click Tree→Export Tree Image→Whole.</li> <li>- After specifying the filename, in the Export Image As dialog, select the Files of type – to TXT files, and, then click &lt; Export &gt;</li> </ul>
17.	Create flowchart of the current node	<ul style="list-style-type: none"> <li>- Click  on the Tree toolbar</li> </ul>
18.	Create CallFlow of the current node	<ul style="list-style-type: none"> <li>- Click  on the Tree toolbar</li> </ul>

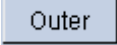
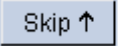
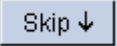



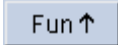
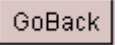
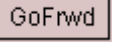
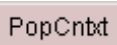




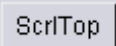
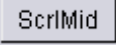
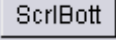
## 6. Browse Window



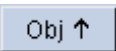
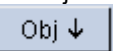






1.	Browse Window	<p>The various cards in the Browse Window will help you to browse at the:</p> <ul style="list-style-type: none"> <li>- project-level</li> <li>- file level</li> <li>- function level</li> </ul> <p>You can print the contents of the Browse Window or export as-a-text-file.</p>
2.	Project Files <Prj Files> card	<p>In the Prj Files card,</p> <ul style="list-style-type: none"> <li>- Try double-click and right-click on a file name</li> <li>- Right click on the root node to add files to the project, edit the include path sequence</li> </ul>
3.	Project Browse <Prj Browse> card	<p>In Prj Browse, you can view the list of all:</p> <ul style="list-style-type: none"> <li>- functions, root functions, recursive functions, conditioned-out functions, globals, structs and unions, etc.</li> </ul> <p>Try double-click and right-click on any node.</p>
	Go to a Function's definition	<ul style="list-style-type: none"> <li>- Double-click on a function name</li> </ul>
	View Recursive functions	<ul style="list-style-type: none"> <li>- Click on "+" to expand the "Recursive Functions" node.</li> </ul>
	Print the Tree or Export as a text file	<ul style="list-style-type: none"> <li>- Right-click on the root node; Click on "Print" or Click on "Export To Text File"</li> </ul>
4.	Project Report <Prj Report> card	<p>In Prj Report, you can view the file-by-file list of:</p> <ul style="list-style-type: none"> <li>- functions, root functions, unused functions, globals, structs and unions, enums</li> <li>- #defines, typedefs, Files #included, unused globals, unused locals</li> </ul>
	View / Hide Specific items	<p>As an example, to view only the Globals in each file:</p> <ul style="list-style-type: none"> <li>- Right-click on the root node; Click on Expand / Collapse Group→Globals</li> </ul> <p>To hide all other groups:</p> <ul style="list-style-type: none"> <li>- Right-click on the root node; Click on Expand / Collapse Group→Globals</li> </ul>
	Print the Tree or Export as a text file	<ul style="list-style-type: none"> <li>- Right-click on the root node; Click on "Print" or Click on "Export To Text File"</li> </ul>















5.	Switch To	<ul style="list-style-type: none"> <li>- Double-click on a Recently Visited Function name</li> <li>- Double-click on an Open file name – to go to that window</li> </ul>
6.	Outline	<p>View the outline of the current file. Go to any function, comment, #include, #ifdef etc.in the file by clicking in the Outline window.</p>
	List of functions in the current file	<ul style="list-style-type: none"> <li>- Click on  icon to view the list of functions in the file. Double-click on a function to go to its definition</li> <li>- Click on other icons to select/deselect them.</li> </ul>
	Use Outline to walk through the file	<p>In Options→Environment→Browse Window, select "Go to Occurrence on Single Click in Outline"</p> <ul style="list-style-type: none"> <li>- Click and select any item in the Outline card of the Browse Window.</li> <li>- Simply press the  key on the keyboard to go to the next item and thus walk through the file.</li> </ul>
	View the #if, #ifdef, etc affect	<p>When you select an #ifdef, etc. in the list in the Outline card:</p> <ul style="list-style-type: none"> <li>- the lower panel shows the scope of the #ifdef and the items that are affected by it.</li> </ul>
7.	File Report	View a function and data summary of the current file. Double-click to go to the corresponding line in the file.
8.	Flowchart	The flowchart card displays a condensed view of the Flowchart.
	View the conditions that must be true	Select < Show/Hide Outer Window >  in the toolbar to display the list of conditions enclosing the current flowchart symbol.
9.	Function's outline	<p>The Function Outline / Area Outline display's a skeletal outline of the current function or Class/Namespace. It is useful for navigating in long areas. You can also view the sequence of all the #if, #ifdef, #else etc. in the file.</p>
	View the conditions that must be true	In the lower part you can see the sequence of conditions - if, else, for, while etc. that must be true in order to reach the current line in the File window
10.	Function Properties/Area Properties	<p>View and go to:</p> <ul style="list-style-type: none"> <li>- Callers of the current function, Functions called by the current function</li> <li>- Structures, #defines, globals etc. used by the current function</li> </ul> <p>Try single-click and double-click on the nodes.</p>
	Callers of the current function	<p>Expand the "Called By " node</p> <ul style="list-style-type: none"> <li>- A double-click on a function name will take you to the definition of that function.</li> </ul>

## 7. Navigation

1.	Display Quick Buttons	Click < Make the File Window Taller > icon at the bottom of the Crystal Window. Or click pull-down menu View→Tall Window
2.	Outer indent	Click  to move the cursor to the outer indent level.
3.	Skip Deeper Indents	Click any one -   . For example: <ul style="list-style-type: none"> <li>- to go from a function to the previous / next function</li> <li>- skip the body of a loop</li> <li>- to go from a case to the previous / next case</li> </ul>
4.	Sentence wise navigation	Click  or  buttons to go to the next or previous sentence. You can skip a declaration with a very long initializer – with just a single click.
5.	Function wise navigation	Click  or  to go to the next or previous function
6.	Go back and Go forward	To go back to the just previous position - you were editing at – click  To go back to the point where you clicked the previous < Go Back >, click 
7.	Pop Context	Click  to return to the file-and-location preceding the most recent browse operation.
8.	Go to Modified lines	Visit the lines of code that were modified by you by clicking  and  . Crystal REVS for C and C++ remembers modified lines even across sessions.
9.	Go to Marked lines	To visit the marked lines: click  and  .
10.	Scroll half a page	<ul style="list-style-type: none"> <li>- Click  to bring the current line to the top of the screen.</li> <li>- Click  to bring the current line to the middle of the screen.</li> <li>- Click  to bring the current line to the bottom of the screen.</li> </ul>



11.	Next or Previous word	<p>To search for a whole word:</p> <ul style="list-style-type: none"> <li>- Place the cursor on a function name or identifier</li> <li>- Click the  or </li> </ul>
12.	Next or Previous object	<p>To search for an object as per the C/C++ scope rules:</p> <ul style="list-style-type: none"> <li>- Place the cursor on the object</li> <li>- Click  or </li> </ul>
13.	Find and Replace	
	Plain Text Search / Object Search	You can use the search-panel in the search toolbar or the Find dialog.
	Lexical Search	<p>To search for a sequence of tokens as per C/C++ lexical rules (ignore differences in amount of white space, match across line boundaries)</p> <ul style="list-style-type: none"> <li>- Click the  in the <a href="#">search toolbar</a> or press  +  to bring the Find dialog.</li> <li>- Set the mode to Lexical Search</li> <li>- Enter the search string in the "Search for" panel.</li> <li>- Set the Scope – File/Function/Selected text.</li> <li>- Click &lt; Find &gt;</li> </ul>
	Regular Expression Search	<p>To search a regular expression</p> <ul style="list-style-type: none"> <li>- Click  in the <a href="#">search toolbar</a> or press  +  to bring the Find dialog.</li> <li>- Set the mode to Regular Expression Search</li> <li>- Enter the regular expression in the "Search for" panel.</li> <li>- Set the Scope – File/Function/Selected text.</li> <li>- Click &lt; Find &gt;</li> </ul>

	Mark Lines	<p>To mark all lines that contain a matching string</p> <ul style="list-style-type: none"> <li>- Click  in the search toolbar_or press  +  to bring the Find dialog.</li> <li>- Set the search mode</li> <li>- Enter the search string</li> <li>- Set the Scope – File/Function/Selected text.</li> <li>- Click &lt; Mark All &gt;.</li> </ul>
14.	Find in Files (GREG)	<p>Press  +  or click on the  in the search toolbar to display the Find in Files dialog.</p> <ul style="list-style-type: none"> <li>- Set the Search Mode: Text, Lexical, Regular Expression, Object, Modified Object.</li> <li>- Specify the search string into “Search For ” panel.</li> <li>- Specify the target folders: You can select one of CURRENT FILE, DESKTOP FILES, or PROJECT FILES, or, use the Browse button and select the folder.</li> <li>- Click &lt; Perform &gt; button.</li> </ul> <p>The search results are displayed in the output window under the GREG card.</p> <p>You can view the results as a list or as a tree with </p>
	Where modified occurrences	<p>Modified occurrences of an object are those occurrences where the object is being used on the left hand side of an assignment, or, contents of the object are being modified by passing the object as a reference to a function, or, the object is being subjected to pre or post increment or decrements.</p> <p>To find all such occurrences of an object in the project</p> <ul style="list-style-type: none"> <li>- Place the cursor on the desired identifier.</li> <li>- Right-click, then click “Where Modified Occurrences”.</li> </ul>
	GREG and Replace	<p>In the output window, in the GREG card</p> <ul style="list-style-type: none"> <li>- Click  to bring the Replace dialog.</li> <li>- Specify the Replace By text.</li> <li>- Check/Uncheck the desired File Modification items.</li> <li>- Specify one of the Replace Occurrences option – Current”, “All in current file” and “In all files”.</li> <li>- Click &lt; Replace &gt;.</li> </ul>

15.	Go to Definition, Declaration and Prototype	<ul style="list-style-type: none"> <li>- Right click in the File Window, then click &lt; ShowDefinition &gt; to go to any object's definition.</li> <li>- Right click in the File Window, then click &lt; ShowDeclaration &gt; to go to any object's nearest declaration.</li> <li>- Place the cursor on a function name and click Goto→Prototypes to view a function's prototype</li> </ul>
16.	Perfect pair match	<p>When the cursor is at one of: (, [, {, /* and you click , the cursor moves to the corresponding ), ], }, */ and vice versa.</p> <p>When matching braces, i.e. { and }, it is not necessary to place the cursor exactly at the brace.</p> <p>You can place the cursor anywhere on the line</p>
17.	From a #if, #ifdef to matching #else, #endif, and vice-versa	<p>When the cursor is on a #if, #ifdef, #ifndef, click  to go to the matching #else, #elif or #endif.</p>
18.	Viewing undeclared object	<p>Click on  to move the cursor downward to an occurrence of an object that is undeclared and not #defined either.</p>
19.	Viewing unused object	<p>Clicking on  moves the cursor to the declaration of a local object that has not been used in the program.</p>
20.	Viewing unmatched braces	<p>Click on Goto→Find Errors→Unmatched else, (, /* etc to move the cursor downward</p> <ul style="list-style-type: none"> <li>- to an unmatched brace i.e. "{“ or ”}",</li> <li>- an unmatched comment delimiter i.e. “/*” or “*/”, or,</li> <li>- an unmatched “else” i.e. an “else” for which there is no “if”.</li> </ul>
21.	Viewing syntax errors	<p>Clicking on Goto→Find Errors→Next Syntax Error moves the cursor downward to the next Syntactically Incorrect C/C++ sentence (if one is present).</p> <p>With this operation you can find syntax errors in a program (even without having to compile it)</p>
22.	Outline Margin	<p>Click Tools→ Show Fold Level Margin to view the Outline Margin for every function within the file.</p> <ul style="list-style-type: none"> <li>- Click on '-' to collapse the current level in the function.</li> <li>- Click on '+' to expand a level in the function.</li> <li>- Right click on '+' to expand all levels under the current level.</li> </ul>

## 8. Rich Trees

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

1.	Create a Rich Tree	<ol style="list-style-type: none"><li>1. Create the desired Call-tree or Caller-tree</li><li>2. Click &lt; Search in Tree &gt;  in the Tree window toolbar to open the Search in Tree dialog.</li><li>3. In the Enable Display card, select the additional information that should be shown with each function node.</li><li>4. Click &lt; Apply &gt;</li><li>5. Also use the "Object Search" card to search for specific data objects and display them in the relevant tree nodes</li></ol>
2.	The "Search in Tree" dialog	Click the < Search in Tree >  in the Tree Window toolbar to view the "Search in Tree dialog".
	The Enable Display card	Use this card to: <ul style="list-style-type: none"><li>– Enable/disable information that is displayed alongside each node</li></ul>
	The Highlight Occurrences card	Use this card to: <ul style="list-style-type: none"><li>– Select the criterion by which other occurrences of a node are highlighted</li></ul>
	The Object Search card	In this card, <ol style="list-style-type: none"><li>1. Specify the data object or struct name, etc</li><li>2. Find in what functions the data object is used - by clicking &lt; Search &gt; or &lt; Search All &gt;</li></ol>
	The Advanced Search card	Using this card, you can: <ul style="list-style-type: none"><li>– find the next occurrence of a function in the tree.</li><li>– find the next function used from a specific file.</li><li>– use the "history list" to search for a particular object, struct, class, type or #define</li><li>– search for a string in all the text that is being displayed in the tree-nodes..</li></ul>
3.	Attach information to tree nodes	Use the "Enable Display" card in the "Search in Tree" dialog
	View parameters of functions	Select "Function Parameter List"; click < Apply >.
	View stack depth at each node	Select "Stack Size"; click < Apply >.

	File Names where Functions are defined	Select "File Names"; click < Apply >.
	Attach Comments to tree nodes	At each tree-node, view the comment associated with the function call: Select "Associated Comment "; click < Apply >.
	Objects used by functions	Select "Object Name"; click < Apply >. Use the "Object Search" card to search for specific data objects and display them in the relevant tree nodes
4.	Which functions use a category of objects	Use the "Object Search" card of the Search in Tree dialog
	View list of globals used by each function	<ul style="list-style-type: none"> <li>– Set "Type of Object" to &lt;Globals&gt;</li> <li>– Set "Select From" to &lt;Any Global&gt;.</li> </ul> At this point, "Object Name" contains < Any Global > <ul style="list-style-type: none"> <li>– Click &lt; Search &gt;</li> </ul>
	List of Structures used by each function	<ul style="list-style-type: none"> <li>– Set "Type of Object" to &lt;Structures/Unions&gt;</li> <li>– Set "Select From" to &lt;Any Structure/Any Union&gt;.</li> </ul> Now "Object Name" contains < Any Structure/Any Union >. <ul style="list-style-type: none"> <li>– Click &lt; Search &gt;</li> </ul>
	Which members of a Structure are used by the functions	<ul style="list-style-type: none"> <li>– Set "Type of Object" to &lt; Structures/Unions &gt;</li> <li>– In "Select From", select the name of the structure you are interested in</li> </ul> At this point, "Object Name" contains the name of the structure <ul style="list-style-type: none"> <li>– From 'Members' Click on the dot or arrow i.e. "." or "-&gt;"</li> <li>– Set "Select From" to &lt;Any Member&gt;.</li> </ul> Now "Object Name" contains: name-of-the-desired-structure.< Any Member > <ul style="list-style-type: none"> <li>– Click &lt; Search &gt;</li> </ul>
5.	Which functions use a specific object :	Use the "Object Search" card of the Search in Tree dialog




	Which functions use a specific global	<ul style="list-style-type: none"> <li>– Set "Type of Object" to &lt;Globals&gt;</li> <li>– In "Select From", use the drop-down list to select the name of the global variable you are interested in</li> </ul> <p>Now "Object Name" contains the name of the global variable</p> <ul style="list-style-type: none"> <li>– Click &lt; Search &gt;</li> </ul>
	Which functions use a specific member of a specific structure variable	<ul style="list-style-type: none"> <li>– Set "Type of Object" to &lt; Structures/Unions &gt;</li> <li>– In "Select From", select the name of the structure variable you are interested in</li> </ul> <p>At this point, "Object Name" contains the name of the structure variable</p> <ul style="list-style-type: none"> <li>– From 'Members' Click on the dot or arrow i.e. "." or "-&gt;"</li> <li>– In "Select From", select the name of the specific member you are interested in</li> </ul> <p>Now "Object Name" contains: name-of-the-desired-structure. name-of-the-desired-member</p> <ul style="list-style-type: none"> <li>– Click &lt; Search &gt;</li> </ul>
6.	Advanced Search	In the "Advanced Search" card of the "Search in Tree" dialog:
	Go to next node that is from a specific file	<ul style="list-style-type: none"> <li>– Select the "File Name" checkbox. (the "Search String" check box should be OFF)</li> <li>– Use the drop-down list to select the filename you are interested in.</li> <li>– Click &lt; Search &gt; to find the next tree node that is from the selected file; Click &lt; Search All &gt; to find all nodes that are from the selected file.</li> </ul>
	Go to the next node that uses a specific object	<ul style="list-style-type: none"> <li>– Select the "Object Name" checkbox. (the "Search String" check box should be OFF)</li> <li>– Use the history list to select the object you are interested in.</li> <li>– Click &lt; Search &gt; to find the next node containing the object name; Click &lt; Search All &gt; to find all nodes containing a match for the selected object.</li> </ul>
	Search for a string in tree-nodes	<ul style="list-style-type: none"> <li>– Select the "Search String" checkbox.</li> <li>– Enter the name of the string you wish to search for.</li> <li>– Click &lt; Search &gt; to find the next node containing the string; Click &lt; Search All &gt; to find all nodes containing a match for the selected string.</li> </ul>
7.	Highlight other Occurrences	Use the "Enable Display" card in the "Search in Tree" dialog
	Highlight nodes that have the same function name as the current node	Select "Function Name"; click <Apply>.









	Highlight nodes that are from the same file as the current node	Select "File Name"; click <Apply>.
	Highlight nodes that has the same object-name attached as the current node	Select "Object Name"; click <Apply>.
8.	Single-click to go to the function-call	<p>A single click on a node in the Call-Tree:</p> <ul style="list-style-type: none"> <li>– moves the cursor to the first call of the function in the parent's code</li> <li>– Subsequent clicks will move the cursor to the next call of the function in the parent's code</li> </ul>
9.	View the DataFlow of the data object	<ul style="list-style-type: none"> <li>– Click to select a tree-node that has a data-object displayed alongside it</li> <li>– Right-click anywhere in the Tree Window, then click &lt; DataFlow for the Current Object &gt; in the pop-up menu.</li> </ul>
10.	Filtering the tree	After searching for a specific function in the tree, or when information is attached to the tree nodes, you can filter the tree so that only the nodes of interest are visible and the rest of the tree becomes hidden.
	View only the nodes that have objects	<p>After you search for an object and, as a result, that object is displayed alongside a few nodes in the tree:</p> <ul style="list-style-type: none"> <li>– Right click anywhere in the Tree Window.</li> <li>– Click &lt; Show All Paths for Object &gt;.</li> </ul> <p>Now only the sub-trees that contain the nodes-with-objects are visible, the rest of the tree becomes hidden.</p>
	View nodes that are from a specific file	<p>After you enable display of filenames, so that filenames are displayed alongside the tree-nodes:</p> <ol style="list-style-type: none"> <li>1. Select a node that contains the filename of interest.</li> <li>2. Right click anywhere in the Tree Window.</li> <li>3. Click &lt; Show All Paths for Filename &gt;.</li> </ol> <p>Now the sub-trees that contain the nodes-with-filename of interest are visible, the rest of the tree becomes hidden.</p>
	Search ALL	<p><b>How "Search ALL" works:</b> Crystal will search to find all the nodes based on the search criterion.</p> <p>If any unexpanded parts of the tree contain search-matches, those parts will be automatically expanded.</p> <p>Only the sub-trees that contain search-matches remain visible, the rest of the tree becomes hidden.</p>





	Show Whole Tree	<p>When a few sub-trees are visible and the rest of the tree is hidden, new searches and a few other operations <u>can not be performed</u>.</p> <p>To display the whole tree again:</p> <ul style="list-style-type: none"><li>– Right click anywhere in the Tree Window.</li><li>– Click &lt; Show Whole Tree &gt;.</li></ul>
--	-----------------	--



## 9. Data Flow

1.	Create a DataFlow	<ul style="list-style-type: none"> <li>Place the cursor on an object name in the file window.</li> <li>Click on the &lt; Create DataFlow &gt;  in the toolbar.</li> </ul>
	What does DataFlow show?	<ul style="list-style-type: none"> <li>The statements that use the data object, the function-calls that take the data object as an argument</li> <li>The control flow around the use of data object</li> </ul>
	DataFlow enables you to:	<ul style="list-style-type: none"> <li>Quickly find all the functions that use the data object,</li> <li>How they use the data object – get a complete view of data use</li> <li>When passed as an argument, expand the function-call - to view how it is used in the called function</li> <li>With the graphical view, you can understand and remember much faster.</li> </ul>
2.	DataFlow of a local	<p>A local variable is known only within a function.</p> <ul style="list-style-type: none"> <li>When you create the DataFlow of a local variable, you will see how it is used in the current function.</li> <li>When passed as an argument, expand the function-call - to view how it is used in the called function</li> </ul>
3.	DataFlow of a global	<p>A global variable is visible to more than one function:.</p> <ul style="list-style-type: none"> <li>In the DataFlow for a global, you will see all the functions where the global is used.</li> <li>Double-click and expand any of those functions to view the use of the global in that function.</li> </ul>
4.	Dataflow of instances of a struct	<p>DataFlow of a struct member in <b>all the instances</b> of the struct</p> <ul style="list-style-type: none"> <li>Go to the member declaration and place the cursor on the member's name.</li> <li>Click on the &lt; Create DataFlow &gt;  in the toolbar</li> </ul>
	Dataflow of an instance of a struct	<p>DataFlow of a struct member in a <b>particular instance</b> of the struct</p> <ul style="list-style-type: none"> <li>Go to the code where it is used as member of that particular instance.</li> <li>Place the cursor on the member's name.</li> <li>Click on the &lt; Create DataFlow &gt;  in the toolbar</li> </ul>


	Dataflow of all instances of a struct	<p>DataFlow of all instances of a struct</p> <ul style="list-style-type: none"> <li>– Place the cursor on the structure name i.e. the struct tag.</li> <li>– Click on the &lt; Create DataFlow &gt;  in the toolbar</li> </ul>
5.	Expand function-call in a DataFlow	<p>The presence of a function-call involving the data object is indicated by the  icon above the symbol.</p> <ul style="list-style-type: none"> <li>– Double-click on a symbol that contains a function call.</li> </ul> <p>The expansion is inserted just before the function call.</p>
	Collapse an expansion	Double-click on the start symbol of the expansion
6.	View the code covered by miniature symbols	<ul style="list-style-type: none"> <li>– Click on miniature symbols and view the code covered by them.</li> </ul>
	Scroll highlighted code	<ul style="list-style-type: none"> <li>– Click the &lt; Scroll Code Up &gt;  and &lt; Scroll Code Down &gt;  in the Flowchart window toolbar</li> </ul>
7.	Easily navigate to many functions in many files	<p>Click on a flowchart symbol and view the corresponding code in the Source File</p> <ul style="list-style-type: none"> <li>– Click on the symbol containing “Tracking: ....” to go to the start of the function in file window.</li> <li>– In a DataFlow expansion, click on the start symbol to view the function-call in the file window</li> </ul>
8.	View the CallStack	<p>When you expand function-calls in a DataFlow, often you have a sequence of DataFlows that are nested in one another:</p> <p>To view the sequence of functions that brought you to a nested expansion:</p> <ul style="list-style-type: none"> <li>– Right-click in any symbol in the nested expansion</li> <li>– Select &lt; Function Call Stack &gt; in the pop-up menu</li> </ul> <p>Now you can see the full call stack.</p> <ul style="list-style-type: none"> <li>– Click on a function’s-name in the call-stack to go to its expansion</li> </ul>
9.	View the function’s full flowchart	<ul style="list-style-type: none"> <li>– Click on any symbol in the DataFlow so that the corresponding code is highlighted in the File Window.</li> <li>– Click  in the main toolbar at the top.</li> </ul>
	Go back to DataFlow	<p>To go back to the DataFlow:</p> <ul style="list-style-type: none"> <li>– Click &lt; Back to Flowchart &gt;  in the Flowchart toolbar</li> </ul>


## 10. Call Flow

1.	Create a CallFlow	<ul style="list-style-type: none"> <li>– Place the cursor anywhere in the function.</li> <li>– Click the &lt; Create CallFlow &gt;  in the toolbar.</li> </ul>
2.	What do you get with the CallFlow	<p>You get a simplified view of the function. You can easily see:</p> <ul style="list-style-type: none"> <li>– the functions called in that function</li> <li>– the control flow of the function</li> </ul>
3.	View the code covered by miniature symbols	<ul style="list-style-type: none"> <li>– Click on miniature symbols and view the code covered by them.</li> </ul>
	Scroll highlighted code	<ul style="list-style-type: none"> <li>– Click the &lt; Scroll Code Up &gt;  and &lt; Scroll Code Down &gt; .</li> </ul>
4.	CallFlow enables you to:	<ul style="list-style-type: none"> <li>– Make a fast attack on the current function and its called-functions</li> <li>– Understand a function's logic in detail</li> </ul>
5.	Expand function-calls in a CallFlow	<p>In CallFlow, the presence of a function-call is indicated by the  icon above the symbol.</p> <ul style="list-style-type: none"> <li>– Double-click on a symbol that contains a function call.</li> </ul> <p>The expansion is inserted just before the function call.</p>
	Collapse an expansion	<ul style="list-style-type: none"> <li>– Double-click on the start symbol of the expansion</li> </ul>
6.	Easily navigate to many functions in many files	<p>Click on a flowchart symbol and view the corresponding code in the Source File</p> <ul style="list-style-type: none"> <li>– Click on the symbol containing “Tracking: ....” to go to the start of the function in file window.</li> <li>– In a CallFlow expansion, click on the start symbol to view the function-call in the file window</li> </ul>

7.	View the CallStack	<p>When you expand function-calls in a CallFlow, often you have a sequence of CallFlows that are nested in one another:</p> <p>To view the sequence of functions that brought you to a nested expansion:</p> <ul style="list-style-type: none"> <li>– Right-click in any symbol in the nested expansion</li> <li>– Select &lt; Function Call Stack &gt; in the pop-up menu</li> </ul> <p>Now you can see the full call stack.</p> <ul style="list-style-type: none"> <li>– Click on a function's-name in the call-stack to go to its expansion</li> </ul>
8.	View the function's full flowchart	<ul style="list-style-type: none"> <li>– Click on any symbol in the CallFlow so that the corresponding code is highlighted in the File Window.</li> <li>– Click  in the main toolbar at the top.</li> </ul>
	<a href="#">Go back to CallFlow</a>	<p>To go back to the CallFlow:</p> <ul style="list-style-type: none"> <li>– Click &lt; Back to Flowchart &gt;  in the Flowchart toolbar</li> </ul>

## 11. Data-Dependency Trees

1.	Create a Data-Dependency Tree	<p>Place the cursor on any data-object.</p> <ul style="list-style-type: none"> <li>Click &lt;Create Data Dependency Tree&gt;  icon in the toolbar.</li> </ul>
	Data-Dependency Tree of a global	<p>Place the cursor on a global and create it's data-dependency tree.</p> <p>The data-dependency tree of a global variable will contain</p> <ul style="list-style-type: none"> <li>All occurrences where the global variable is being modified</li> <li>If a global variable is being passed as a function argument, then the name of the function.</li> <li>All modified occurrences of the objects modifying the global variable; and so on.</li> </ul>
	Data-Dependency Tree of a local	<p>Place the cursor on a local and create it's data-dependency tree.</p>
	Data-Dependency Tree of a member	<p>Place the cursor on the structure member and create it's data-dependency tree.</p> <p>The data-dependency tree will contain:</p> <ul style="list-style-type: none"> <li>All modified occurrences of the member.</li> <li>All modified occurrences of the objects modifying the member</li> <li>If the member is being passed as an argument, then the name of the function.</li> </ul>
2.	Expand the Data-Dependency tree	<p>Expand the nodes of the Data-dependency tree by clicking on “+”</p>
	Expand the whole Tree	<p>To <b>fully expand</b> the sub-tree under the current node, Right click anywhere in the Tree window; Click &lt; Expand Full &gt;</p>
3.	Navigating	<p>Double-click on any node in the Data-Dependency tree to go to the corresponding source code line.</p>

4.	Node Prefixes	<ul style="list-style-type: none"> <li>- <b>A</b> - indicates that the node is an array</li> <li>- <b>D</b> - indicates that the node is a defined constant</li> <li>- <b>E</b> - indicates that the node is an enum variable</li> <li>- <b>F</b> - indicates that the node is a function</li> <li>- <b>G</b> - indicates that the node is a global</li> <li>- <b>M</b> - indicates that the node is structure or union member</li> <li>- <b>P</b> - indicates that the node is a pointer</li> <li>- <b>T</b> - indicates that the node is a typedef</li> <li>- <b>EC</b> - indicates that the node is an enum constant</li> <li>- <b>FA</b> - indicates that the node is a function argument</li> <li>- <b>FP</b> - indicates that the node is pointer to function</li> <li>- <b>GA</b> - indicates that the node is a global array</li> <li>- <b>MP</b> - indicates that the node is a member pointer</li>   <li>- <b>+</b> - indicates that the node is EXAPANDABLE</li> <li>- <b>-</b> - indicates that the node is EXPANDED</li> <li>- <b>*</b> - indicates that the node is RECURSIVE</li> </ul>
5.	The "Search in Tree" dialog	Click the < Search in Tree >  in the Tree Window toolbar to view the "Search in Tree dialog".
	The Enable Display card	<p>Use this card to:</p> <ul style="list-style-type: none"> <li>- Enable/disable information that is displayed alongside each node</li> </ul>
	The Highlight occurrences card	<p>Use this card to:</p> <ul style="list-style-type: none"> <li>- Select the criterion by which other occurrences of a node are highlighted</li> </ul>
	The Advanced Search card	<p>Using this card, you can:</p> <ul style="list-style-type: none"> <li>- find the next occurrence of an object name</li> <li>- find the next object used from a specific file.</li> <li>- find the next occurrence of a function in the tree.</li> <li>- use the "history list" to search for a particular object, struct, class, type or #define</li> <li>- search for a string in all the text that is being displayed in the tree-nodes..</li> </ul>



6.	Attach information to tree nodes	<p>Each node in the data-dependency tree shows</p> <ol style="list-style-type: none"> <li>1. Name of data object</li> <li>2. function name</li> <li>3. file name</li> <li>4. Line number</li> </ol> <p>Use the “Enable Display” card in the “Search in Tree” dialog</p>
	View function names in which objects are used	Select “Function Name”; click < Apply >.
	File Names where Objects are used	Select “File Names”; click < Apply >.
	Attach Comments to tree nodes	<p>At each tree-node, view the comment associated with the function call:</p> <p>Select "Associated Comment "; click &lt; Apply &gt;.</p>
	Objects used by functions	<p>Select “Object Name”; click &lt; Apply &gt;.</p> <p>Use the "Object Search" card to search for specific data objects and display them in the relevant tree nodes</p>
7.	Search for a Object in the Data Dependency Tree	Use the “Advanced Search” card of the “Search in Tree” dialog
8.	To view the occurrence of an object deep in the Tree	<ul style="list-style-type: none"> <li>– Select the node corresponding to the object</li> <li>– Right click in the tree window, and, then click &lt; Show Path &gt;. Now you can see the path from the root to the node for the object.</li> </ul>

## 12. Generating HTML Documentation

1.	Generate HTML Documentation	<ul style="list-style-type: none"> <li>– Click Project→Create HTML Document</li> <li>– In The HTML Documenter Settings Dialog, select the desired options.</li> <li>– Click &lt; OK &gt;</li> </ul> <p>Crystal C/C++ will begin creating the HTML documentation for the current project.</p> <p>When the documentation completes:</p> <ul style="list-style-type: none"> <li>– Click &lt; OK &gt; - the default document will be displayed in your browser.</li> </ul>
	Open HTML Document	<p>To Open the HTML documentation for the current project:</p> <ul style="list-style-type: none"> <li>– Click Project→Open HTML Document</li> </ul>
	Including Project Specific information	<p>To include project notes, comments, development history, on any specific project related content in the documentation:</p> <ul style="list-style-type: none"> <li>– Click Project→Open Notes – to open the project notes file</li> <li>– Enter the content to be included as part of documentation in the project notes file and save it.</li> <li>– Now create the documentation.</li> </ul>
	Configuring the HTML Documentation Content	<p>Use the HTML Documenter Settings Dialog to configure the contents of the HTML documentation.</p>
2.	HTML Documentation provides	<ul style="list-style-type: none"> <li>– Team-wide viewing of source code</li> <li>– Publish source code on internet and CD-ROM</li> <li>– Full source code browsing in the absence of an IDE.</li> </ul>
	HTML Documents also provide	<ul style="list-style-type: none"> <li>– Comprehensive information – including flowcharts, trees, code-metrics, cross-reference, etc</li> <li>– Easy to browse, easy to comprehend, wide code coverage</li> <li>– Ideal platform to discuss designs with clients and global teams – without actually transferring source code.</li> </ul>
3.	Contents of Crystal HTML Documents	<ul style="list-style-type: none"> <li>– File documentation</li> <li>– File Include-tree</li> <li>– Source code listing - beautified and syntax-highlighted.</li> <li>– Flowcharts for all-functions</li> <li>– Call-tree, caller-tree, etc</li> <li>– Code Metrics</li> <li>– Cross-reference</li> </ul>

		<ul style="list-style-type: none"> <li>– Static Check documentation</li> </ul>
4.	HTML Documenter settings dialog	Configure the contents of the documents here.
	Specifying the documentation folder	<ul style="list-style-type: none"> <li>– Specify the path where the documentation is to be created in “Path for Documents”.</li> </ul>
	Changing the documentation Title	<ul style="list-style-type: none"> <li>– Enter the “Documentation Title” in the Report Title panel</li> </ul>
	Attaching a logo to the documentation	<ul style="list-style-type: none"> <li>– Browse and choose the LOGO file for the documentation.</li> </ul>
	Enable Code-metrics	<p>Under the General section:</p> <ul style="list-style-type: none"> <li>– Check ‘Enable Code Metrics’ to include code metrics as part of documentation</li> </ul>
	Include source code listing	<p>In the Source Code Section:</p> <ul style="list-style-type: none"> <li>– Check ‘Source Code Listing’ - the formatted source code for each file will to be included in the documentation.</li> </ul>
	Include static-check reports	<p>In the Static Check Section:</p> <ul style="list-style-type: none"> <li>– Select "Enable Static Checker" to static check each project file When selected, the ‘MISRA C’ static checking report for the project will be part of the project.</li> </ul>
	Include function flowcharts	<p>In the Flowcharts section:</p> <ul style="list-style-type: none"> <li>– Check Enable Flowchart creation to include flowcharts for all project functions as part of the document.</li> <li>– Check – Thumbnails with Height and Width</li> <li>– Choose the type of the flowchart with the level of the detail.</li> </ul>
	Include various types of trees	Use the Trees section to select the types of trees to be included as part of the HTML documentation. You can select from
5.	Browsing in HTML documentation	<p>The home page for each documentation set is “default.html”. View the project summary here.</p> <p>Then, click Project Overview→Browse Tree to view the Project tree. Use it to navigate in the documentation.</p>
	Project Browse Tree	<ul style="list-style-type: none"> <li>– Expand the nodes to of the tree,</li> </ul>
	File Documentation	<p>View a file’s individual documentation. Use</p> <ul style="list-style-type: none"> <li>– File Report Tree</li> <li>– Documentation</li> <li>– Source Code Listing</li> </ul>

	Code Metrics	<p>The HTML documentation contains the following Volume Metrics:</p> <ul style="list-style-type: none"> <li>— Project Metrics</li> <li>— File Metrics</li> <li>— Function Metrics</li> </ul> <p>You can also view the following Complexity Metrics:</p> <ul style="list-style-type: none"> <li>— Halstead's Complexity Metrics</li> <li>— McCabe's Complexity Metrics</li> </ul>
	Cross Reference	View the list of all globals in the project, along with where the object has been defined and used.
	Static Check Report	View the static check report – on a per file or a per rule basis.
6.	Documentation Menus	Please view the Help contents. ( Html Documentation→Menus... )

## 13. Static Checking

1.	Static Checking Report for a Project	<p>To static-check a project, click Project→Create Static Checker Report.</p> <p>In the Static-checker dialog, select:</p> <ul style="list-style-type: none"> <li>– The Rule Type from the drop-down list</li> <li>– The rules to be checked under the Rule Type</li> <li>– The output report format.</li> <li>– Click &lt; OK &gt;.</li> </ul> <p>The Static Checker progress meter will be displayed</p> <p>As soon as the report generation is complete, the results will be displayed in the selected format.</p>
	Static Checking a File	To static-check a file, click Tools→ Create Static Checker Report.
	Which Static Checking standards are supported	<p>You can choose from:</p> <ul style="list-style-type: none"> <li>– MisraRule1998 - for MISRA-C:1998</li> <li>– MisraRule2004 - for MISRA-C:2004</li> </ul>
2.	Select the Rules to Check	<p>Before you begin static-checking,</p> <ul style="list-style-type: none"> <li>– Select the rules to be checked</li> <li>– Unselect the rules you want to ignore</li> </ul>
3.	The Static Check dialog	Use this dialog and control the generation of static checking reports.
	Settings	Use the settings menu to Import or Export a pre-defined set of rules to be checked.
	Rule Type and Rules List	Chose the standard to check and then select the rules to check/ignore from the rules list
	General Settings	While reporting violations, if Crystal REVS find's a known violation that has been marked in the code for not to be reported, it will not be reported in the output if ' <i>Report all violations ignoring the embedded pragmas</i> ' is unchecked.

	Output formats	<p>The result of a static check violations test can be:</p> <ul style="list-style-type: none"> <li>– Viewed in the Output Window within Crystal C/C++. You can navigate, view and correct these violations</li> <li>– Viewed as HTML reports generated in a specified folder. These HTML reports can be given to the other members of a development team to fix the various violations.</li> </ul>
4.	Viewing the Report	You can view a static report in the “ <i>Static Checker</i> ” tab of the output window.
5.	Excluding known violations from being reported	For violations which are known to exist in a project, you may want to suppress reporting them every time. To do so, mark such violations in the source code.
	Suppressing violations for the whole file	<p>To suppress reporting rules 5.5 and 5.6 of MISRA 2004</p> <ul style="list-style-type: none"> <li>– In the beginning of the file insert <code>/*#pragma disable( MISRA2004, 5.5, 5.6 )*/</code></li> </ul>
	Suppressing violations for a given range	<p>To suppress reporting rules 5.5 and 5.6 of MISRA 2004</p> <ul style="list-style-type: none"> <li>– In the beginning of the range insert <code>/*#pragma disable( MISRA2004, 5.5, 5.6 )*/</code></li> <li>– In the end of the range for ignoring 5.5, insert <code>/*#pragma enable( MISRA2004, 5.5 )*/</code></li> <li>– In the end of the range for ignoring 5.6, insert <code>/*#pragma enable( MISRA2004, 5.6 )*/</code></li> </ul>
	To add a reason while suppressing a violation	<p>To suppress reporting rules 45:2 of MISRA 1998</p> <ul style="list-style-type: none"> <li>– In the beginning of the range insert <code>/*#pragma disable( MISRA1998, 45.2:“This rules has been suppressed.” )*/</code></li> </ul> <p>You can use any string of your choice after the colon ':'</p>
6.	Static Check Reports and HTML Documentation	<p>The static-check report can be part of the HTML documentation.</p> <p>To include the static check report:</p> <ul style="list-style-type: none"> <li>– Click Project→Create HTML document</li> <li>– Enable Static Checker</li> </ul>
7.	Static Check Reports and Command Line mode	<p>The static-check report can also be generated in the command line mode.</p> <p>Please view the Help contents. ( Static Checking→ ... )</p>

## 14. Comment Trail

1.	Software Change Report	The Comment Trail Marker report is the software change report. All software changes under a single Change ID are grouped and tracked together.
2.	Creating a Comment Trail Marker	<ul style="list-style-type: none"> <li>– Click Insert→Comment Trail Marker→New Comment Trail Marker and create one.</li> <li>– Click &lt; Save &gt; to save the created Comment Trail Marker.</li> </ul>
3.	Comment Trail Marker format	A typical comment trail marker looks as follows <b>CTM: ListOverflow</b> JohnD 11/24/2009: <i>pointer increment was missing here</i> <b>ENDCTM</b>
4.	Inserting Comment Trail Markers	<ul style="list-style-type: none"> <li>– Click Insert→Comment Trail Marker→<i>ListOverflow</i></li> </ul> You can select any Comment Trail Marker of your choice.
5.	Viewing Comment Trail Marker Report	<ul style="list-style-type: none"> <li>– Click Tools→Comment Trail Marker Report→<i>ListOverflow</i> to view the list of all comment trail markers under the ID <i>ListOverflow</i></li> <li>– The Comment Trail Marker report will be displayed in the output window in the Reports card..</li> </ul>
6.	View a list of all Comment Trail Markers in the project	In the Browse Window, <ul style="list-style-type: none"> <li>– View the Prj Browse Card</li> <li>– Click and expand the node “<i>Comment Trail Markers</i>”</li> <li>– Click on a Comment Trail Marker to view it’s report.</li> </ul>
7.	Deleting Comment Trail Markers	To remove occurrences of a Comment Trail Marker from the project <ul style="list-style-type: none"> <li>– Click Tools→Remove Comment Trail Marker</li> </ul> Alternately, you can right click on a Comment Trail Marker in the Prj Browse window, or, use the < Remove > icon in the Report window.

## 15. State Transition Diagrams

1.	State Machine in a flowchart	<ul style="list-style-type: none"> <li>– A rectangular border is drawn around the sequence of statements that correspond to a state.</li> <li>– One or more statements in a state can trigger a transition to another state. A transition is shown by drawing an arrow from that statement to the border of destination state.</li> <li>– You can attach a comment or a note to a transition arrow.</li> </ul>
2.	State Machines as a switch-case	Often a state machine is coded in the form of a switch-case
	Expanding and collapsing a switch	<ul style="list-style-type: none"> <li>– Double-click on a switch symbol to collapse the whole switch to a high-level symbol.</li> <li>– Right-click on the high-level symbol, then click &lt; Expand Symbol &gt; in the pop-up menu. The switch will be fully expanded.</li> <li>– Double-click on the high-level symbol to expand the switch by two levels</li> </ul>
3.	Create a State Transition – process	<p>To create a State Transition Diagram:</p> <ul style="list-style-type: none"> <li>– Enclose each state in a rectangular border</li> <li>– Create a State Transition</li> <li>– Change the routing of a State Transition</li> <li>– Provide a comment to a State Transition</li> </ul>
	Enclose each state in a rectangular border	<p>To mark only a few cases as states :</p> <ul style="list-style-type: none"> <li>– Right-click on the desired case symbol.</li> <li>– Click : State Diagrams→Mark as State</li> </ul> <p>In general, you can also right-click on a for, while, do, if symbol and mark the whole body as a state. You can also mark all cases as a state in one go.</p>
	Create a State Transition	<ul style="list-style-type: none"> <li>– Identify statement which causes a transition to another state (trigger statement).</li> <li>– Right-click in the trigger statement symbol, then click: State Diagrams→Create Transition.</li> <li>– In the "Flowchart State Transition" dialog box, specify the resulting state.</li> <li>– Then, provide a comment that will be attached to the transition arrow</li> </ul>



	Change the routing of a State Transition	<ul style="list-style-type: none"> <li>– Right-click on the symbol which contains the trigger statement.</li> <li>– Click: State Diagrams→Transition Route</li> <li>– In the "Flowchart Transition Route" dialog, change the "From" and "To" settings.</li> </ul>
	Provide a comment to a State Transition	<ul style="list-style-type: none"> <li>– Right-click in the symbol that contains the trigger statement.</li> <li>– In the pop-up menu, click : State Diagrams→Add Comment to Transition</li> </ul>
4.	Modifying the comment in a State Transition	<ul style="list-style-type: none"> <li>– Right-click in the comment attached to the State Transition.</li> <li>– Click : State Diagrams→Transition Comment Properties</li> <li>– Provide the new comment there!</li> </ul>
5.	Display the State Transition with a desired detail level	<ul style="list-style-type: none"> <li>– Right-click on the switch symbol, then click: State Diagrams→Collapse All Cases.</li> <li>– Double-click in the switch symbol to collapse the switch.</li> <li>– Double-click in the high-level symbol. The switch is expanded to two-levels of detail.</li> <li>– Double-click in the switch symbol to collapse the switch.</li> <li>– Right-click in the switch symbol, then click Expand Symbol. You get a fully expanded State Transition Diagram.</li> </ul>
6.	To save the State Transition Diagram	<p>To save the State Transition Diagram :</p> <ul style="list-style-type: none"> <li>– Right-click anywhere in the flowchart window.</li> <li>– In the pop-up menu, click: State Diagrams→Save &amp; Close Flowchart.</li> </ul>

## 16. Edit Operations

1.	Insert <b>{ }</b> around single statements	<p>In a file,</p> <ul style="list-style-type: none"> <li>Click Insert→Insert { } around single statements</li> </ul> <p>Crystal C/C++ scans the file for all condition statements like if( ), for( ; ; ), while( ), do( ), etc which are not followed by a pair of '{ }'.</p> <p>It inserts a '{' immediately after the condition statement; It inserts a '}' immediately after the following statement.</p>
2.	Swap YES and NO parts of an if..else ,, statement	<ul style="list-style-type: none"> <li>Place the cursor on an “if( ) statement” or an “else”</li> <li>Click Edit→Logic Change→Swap the parts of if_else_statement</li> </ul>
3.	Change do while to while loop	<ul style="list-style-type: none"> <li>Place the cursor on a “do statement”</li> <li>Click Edit→Logic Change→Change do_while loop to while loop</li> </ul>
4.	Change while loop to do while loop	<ul style="list-style-type: none"> <li>Place the cursor on a “while( ) statement”</li> <li>Click Edit→Logic Change→Change while loop to do..while loop</li> </ul>
5.	Rematch Braces	<ul style="list-style-type: none"> <li>Place the cursor within a function</li> <li>Click Edit→Logic Change→Rematch Braces</li> </ul>
6.	UnComment	<p>To uncomment:</p> <ul style="list-style-type: none"> <li>Place the cursor within a comment block <b>/* */</b></li> <li>Click Edit→Uncomment</li> </ul>
7.	Comment Out with <b>/* */</b>	<ul style="list-style-type: none"> <li>Select a block of text to be commented.</li> <li>Right click, then click Comment Out using <b>/* */</b></li> </ul>
8.	Comment Out with <b>//</b>	<ul style="list-style-type: none"> <li>Select a block of text to be commented.</li> <li>Right click, then click Comment Out using <b>//</b></li> </ul>
9.	Condition Out / Code out	<ul style="list-style-type: none"> <li>Select the block of code to be coded out</li> <li>Right click, then click Condition out</li> </ul>

10.	Enclose in { }	<ul style="list-style-type: none"> <li>– Select the lines to be enclosed in { }</li> <li>– Right click, then click Enclose in { }</li> </ul>
11.	Enclose in ( )	<ul style="list-style-type: none"> <li>– Select the condition to be enclosed in ( )</li> <li>– Right click, then click Enclose in ( )</li> </ul>
12.	Delete a pair of { }, ( ), [ ], etc	<ul style="list-style-type: none"> <li>– Place the cursor within the pair of { }, ( ), [ ], etc that you wish to remove</li> <li>– Click the &lt; DelPair&gt; button</li> </ul>
13.	Move Comments to right	<p>If comments are sprinkled among the statements, you can get an uninterrupted view of the code:</p> <ul style="list-style-type: none"> <li>– Click Edit → Move Comments → Move All Comments to Comment-field.</li> </ul>
14.	Shift a statement Up or Down	<p>In the Tokens Panel,</p> <ul style="list-style-type: none"> <li>– Click &lt;MvSnt ↑&gt; to move the current sentence upward</li> <li>– Click &lt;MvSnt ↓&gt; to move the current sentence downward</li> </ul>
15.	Select the whole function, a while loop, a whole case, etc	<p>Place the cursor in the desired function, or statement</p> <ul style="list-style-type: none"> <li>– Click on the Select Block icon on the main toolbar.</li> <li>– Choose from the drop-down list.</li> </ul>
16.	Multiple Clipboards	<p>Click Edit→Paste→Paste # N to display Multiple Clipboard.</p>

## 17. Formatting for improved readability

1.	Crystal C/C++ Formatter	<ul style="list-style-type: none"><li>– Crystal Flow's formatter is parser-based. It does a lot more than a formatter that uses lexical analysis only.</li><li>– Alignment of type specifiers, declarators among consecutive declarations</li><li>– Alignment of assignment operators among consecutive assignment statements</li><li>– Formatting of long expressions that take up more than one line - <b>based on precedence of operators</b></li><li>– Proper formatting of macro definitions that contain many statements</li><li>– Maintains a comment-field to the right for easy readability</li></ul>
2.	Formatting files in Crystal C/C++	<p>When Crystal C/C++ opens a file for the first time:</p> <ul style="list-style-type: none"><li>– It uses the current format settings to format the file and saves the format settings on a per file basis.</li><li>– When you open the file again, the saved format settings are used for that file.</li></ul>
3.	To modify Format settings	<p>Click Tools → Options → Environment → File Window → Format Settings</p> <ul style="list-style-type: none"><li>– Set the Indent and Tab Size, etc</li><li>– Change the Code Formatting Style</li><li>– Change the Comment Formatting Style</li><li>– Selectively disable formatting of code or comments.</li></ul>
4.	To apply the modified format setting	<p>To apply the new settings to a file that is currently open:</p> <ul style="list-style-type: none"><li>– Click File→Refresh Current File.</li></ul> <p>(Until you Refresh the file, your changes will not be applied to the current file)</p> <p>To apply the new settings to all files in the project</p> <ul style="list-style-type: none"><li>– Click Project→Reparse Project Files→All</li></ul> <p>(Until you Reparse the project, your changes will not be applied to the whole project)</p>
5.	To disable automatic formatting	<p>Click Tools → Options → Environment → File Window → Format Settings</p> <ul style="list-style-type: none"><li>– In the Format Mode, select None.</li><li>– Apply the modified settings to the files as described above.</li></ul>

## 18. Batch Operations

1.	Batch Output of Flowcharts	<ul style="list-style-type: none"><li>– Click BatchOutput→Flowcharts for Project</li><li>– In the displayed dialog, select the type of flowchart to be output.</li><li>– Select the level of flowchart which should be output</li><li>– Select the target format – Visio, Image, etc</li><li>– Select if you want to print the flowcharts too.</li><li>– Select the folder in which the output should be created.</li><li>– Click &lt; OK &gt;</li></ul>
2.	Batch Output of Trees	<ul style="list-style-type: none"><li>– Click BatchOutput→Graphs for Project</li><li>– In the displayed dialog, select the different trees you want to be output.</li><li>– Select the folder in which the various trees should be output.</li><li>– Click &lt; OK &gt;</li></ul>
3.	Batch Printing Files	<ul style="list-style-type: none"><li>– Click BatchOutput→Files for Project</li><li>– Follow the instructions on the dialog and click &lt; OK &gt;</li></ul>
4.	Batch Output Files, Flowcharts and Trees	<ul style="list-style-type: none"><li>– Click BatchOutput→All to output Files, Flowcharts and Trees in a single go.</li></ul>

## 19. Miscellaneous Operations

1.	Preprocessing a File	<p>To preprocess a file:</p> <ul style="list-style-type: none"> <li>– Click Tools→Preprocess.</li> </ul> <p>Crystal FLOW or REVS preprocess the file and displays the preprocessed output in a different file with the same name and “<b>pre</b>” suffixed in the extension.</p>
2.	Code Metrics Reports	<p>To generate the various Code Metrics Report,</p> <ul style="list-style-type: none"> <li>– Click Project→Generate Code Metrics</li> </ul> <p>In the ‘Code Metrics’ dialog,</p> <ul style="list-style-type: none"> <li>– Select the target folder</li> <li>– Select the output format</li> <li>– Set the color coding – for high, medium and low values</li> <li>– Click &lt; OK &gt;</li> </ul> <p>Please view the Help contents. (Html Documentation-&gt;Browsing in the ....-&gt;Code Metrics)</p>
3.	Publish Source Code	<ul style="list-style-type: none"> <li>– Click Project→Publish Source Code</li> <li>– In the ‘Source Code Publisher’ dialog, browse and select the destination folder</li> <li>– Click &lt; OK &gt;</li> </ul>
4.	Compile a File	<p>To Compile the current file:</p> <ul style="list-style-type: none"> <li>– Click Tools→Compile</li> </ul> <p>The errors and warning messages are displayed in the output window under the compile tab. To select your compiler, settings, preprocessor directives, etc use the Options menu.</p>
	Compile All	<p>To compile all files:</p> <ul style="list-style-type: none"> <li>– Click Tools→Compile All. The Compile Files dialog is displayed.</li> <li>– Select the files to be compiled.</li> <li>– Click &lt; Compile &gt;</li> </ul> <p>The errors and warning messages are displayed in the output window under the compile tab.</p>
5.	Running Batch Files	<p>You can run any batch file from the Crystal C/C++ environment. To run a batch file:</p> <ul style="list-style-type: none"> <li>– Click Tools→Run User Batch File.</li> <li>– In the dialog, specify the Batch File to be executed</li> <li>– Click &lt; Run &gt;</li> </ul>

## 20. Trouble Shooting

For any support/trouble shooting/FAQ or any other queries, contact the 24x7 customer care at [support@sgvsarc.com](mailto:support@sgvsarc.com)

(This page is intentionally blank)