

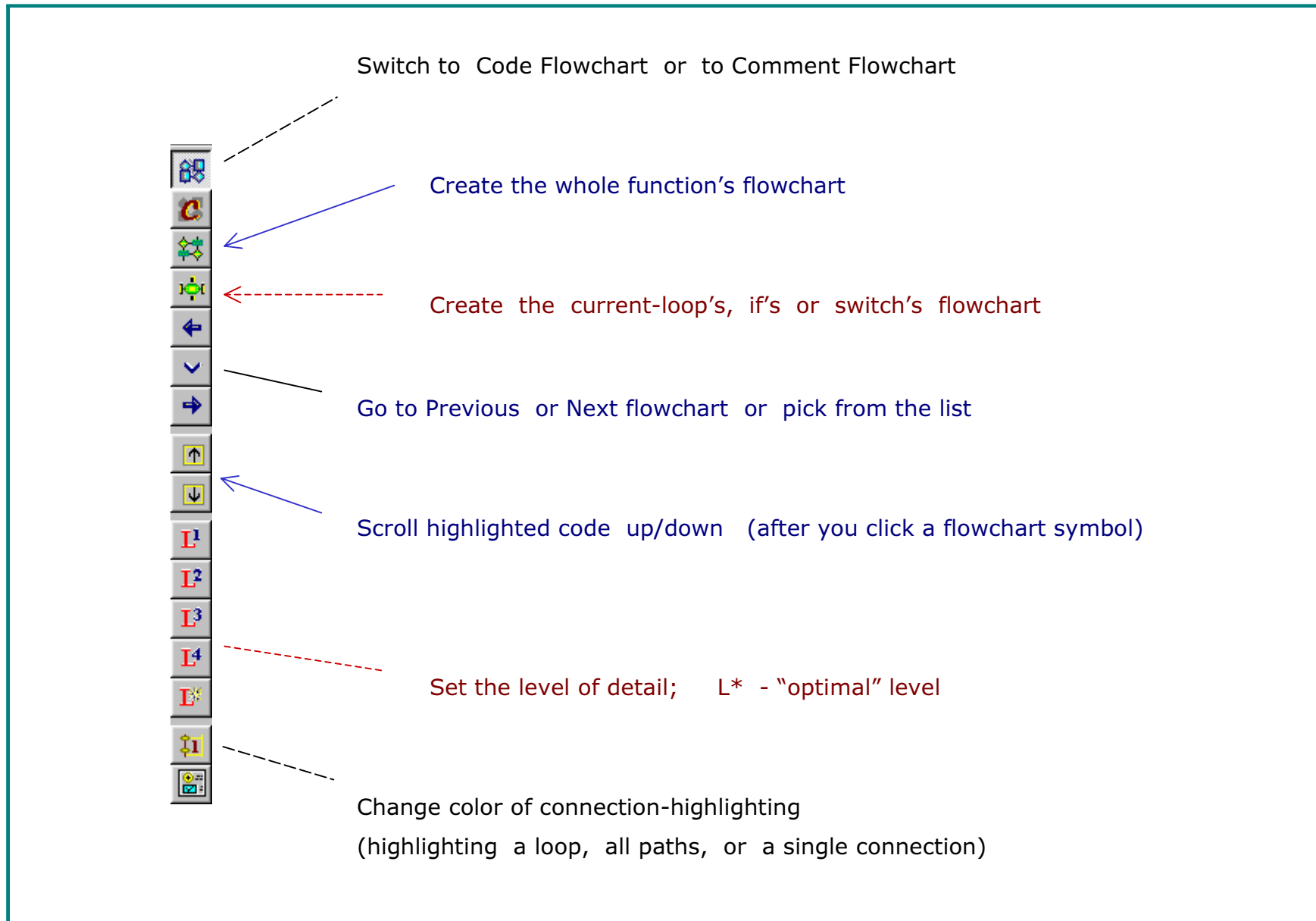
## Viewing Flowcharts in Crystal C/C++

Part 1: Simple Flowcharts

Part 2: Simplify a Complex Flowchart

Part 3: Examples

## Flowchart Toolbar



## Part 1: Simple Flowcharts

1. Condensed View & Detailed View
2. Simple Flowcharts / Complex Flowcharts
3. Bracketing a Loop or an If
4. Highlight One or More Paths
5. Consecutive Nested if's
6. Side-by-side View of the Flowchart & Code
7. View Object's Type

## Flowcharts are Easy to Read with Condensed View & Detailed View

**Tracking Rectangle**

← A flowchart of a simple 60-line function.

- Press the <Home> key to go to the start.
- Press the **cursor keys** ↓, →, ←, ↑ to move through the flowchart.

A **cursor key** brings the next symbol in.  
If the next symbol is far away,  
then for a smaller movement:

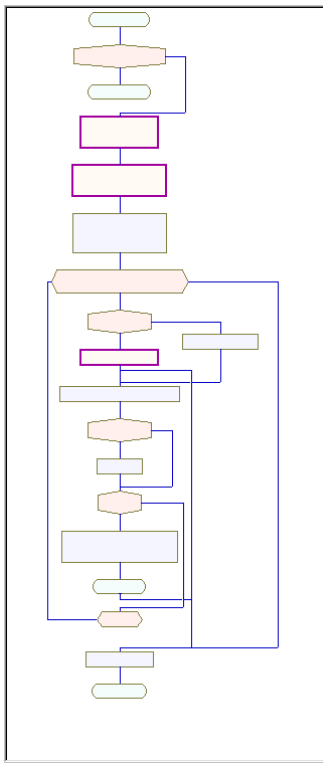
- use **scroll bar** in the detailed view.
- Or move the **tracking rectangle**.

**Condensed View**                      **Detailed View**

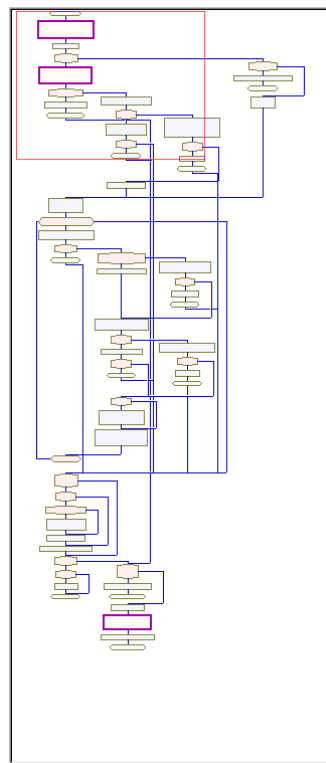
**Tracking Rectangle** indicates the contents of **detailed view**.

## Simple Flowcharts / Complex Flowcharts

### Condensed views of two different functions



A Simple Flowchart



A Complex Flowchart

- ◆ When the condensed view is **simple**, you can easily walk through the detailed view.
- ◆ Sometimes the condensed view is **complex** because
  - the function is very long, and the resulting condensed view is crowded.
  - the function contains many goto's, and so the logic flow is hard to track.

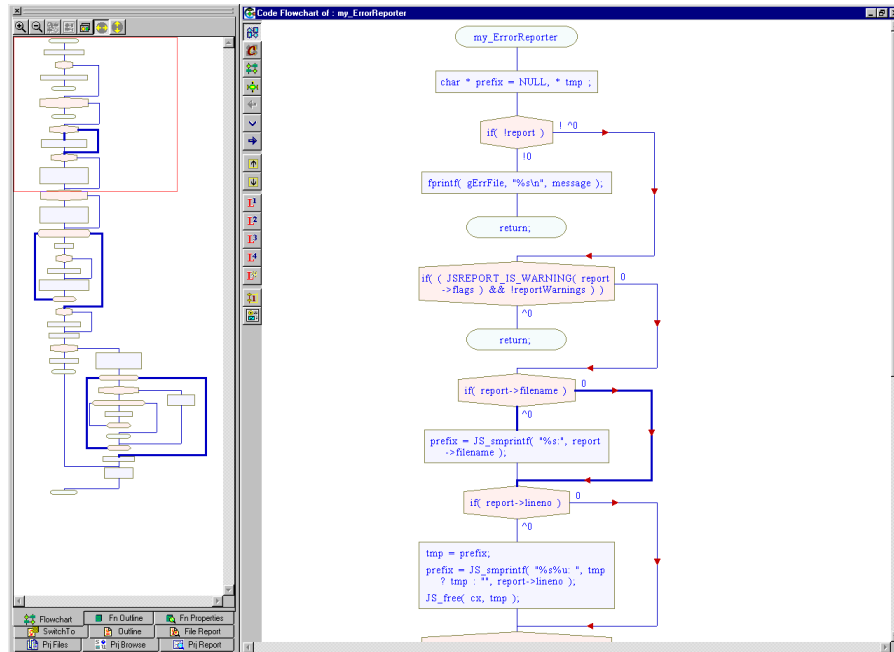
## To Read Simple Flowcharts

When you are going through a simple flowchart  
( or a simplified form of a complex flowchart),

use the following operations:

- ◆ Bracket a loop or bracket an if-else segment of the flowchart.
- ◆ Highlight all paths that can reach a given point in the flowchart.  
Highlight a connection so that it stands out from other neighboring connections.
- ◆ View the type information of all objects that appear in a given symbol.
- ◆ Get a side-by-side view of the flowchart and corresponding code.

## Bracketing a Loop or an If



To bracket a loop or an if statement:

Press the **<Alt>** key and click on a while, do, for, or if symbol in the detailed view.

For additive bracketing, press the **<Ctrl>** key instead of **<Alt>**.

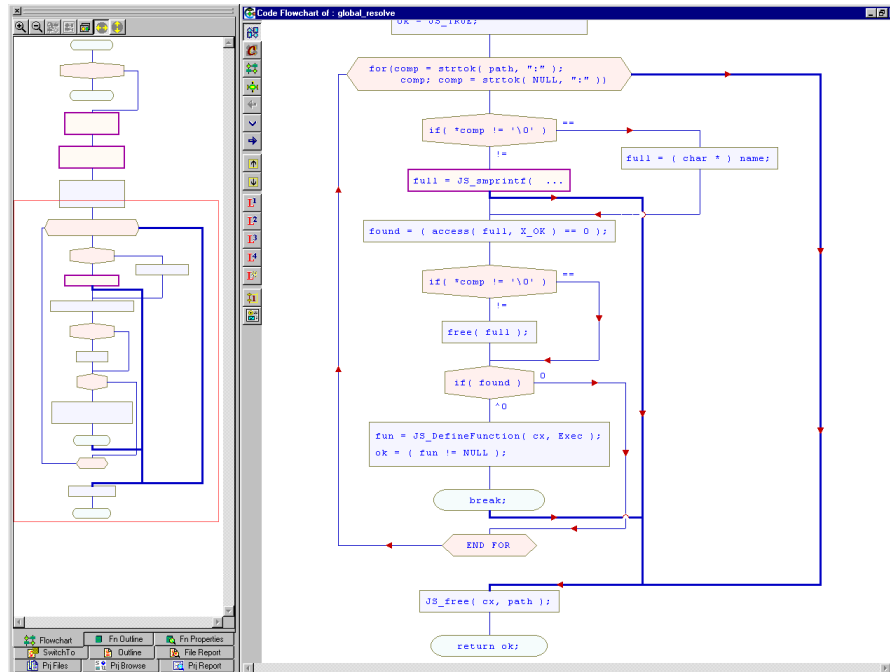
Bracketing is useful for:

- highlighting one or more loops.
- Create visual markers\* in a monotonous flowchart.

\*The above flowchart has a monotonous sequence of if-statements.

Bracket every third if-statement to create visual distinction.

## Highlight One or More Paths



To highlight one or more connections:

- Click on a connection line

When a "break" or "goto" connection crosses over other connections, click on that connection to highlight it. It will help you track the other connections easily.

- Click at the input of a symbol

All paths that can reach the input become highlighted.

- Click at the output of a symbol

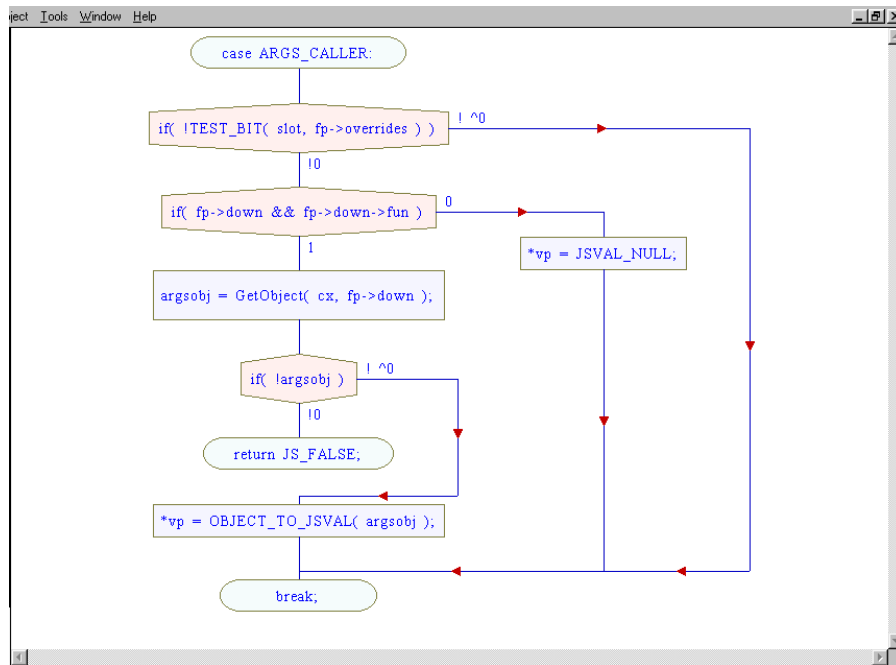
In case of a high-level symbol, it highlights all output branches of that symbol.

For additive highlighting,

press <Ctrl> while clicking as described above.



## Consecutive Nested if's



You can change the labeling style.  
(in the Flowchart card of  
the Customize card in the Options menu)

When there are consecutive nested if's,

- first read downward through all the "YES" branches,
- then read the "NO" i.e. else branches; inner else first, then the outer else.

Labeling of if-symbols:

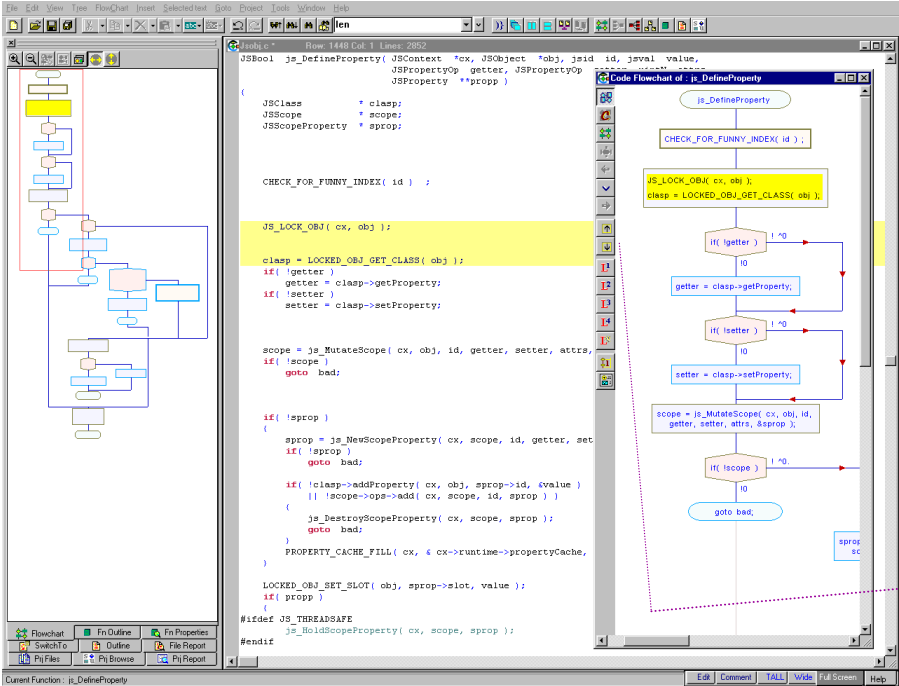
Consider the if-symbol

"if( !TEST\_BIT( slot, fp → overrides ) )"


!0 is analogous to !TEST\_BIT  
i.e. the result of TEST\_BIT is zero.

!^0 indicates TEST\_BIT is non-zero.

## Side-by-side View of the Flowchart & Code

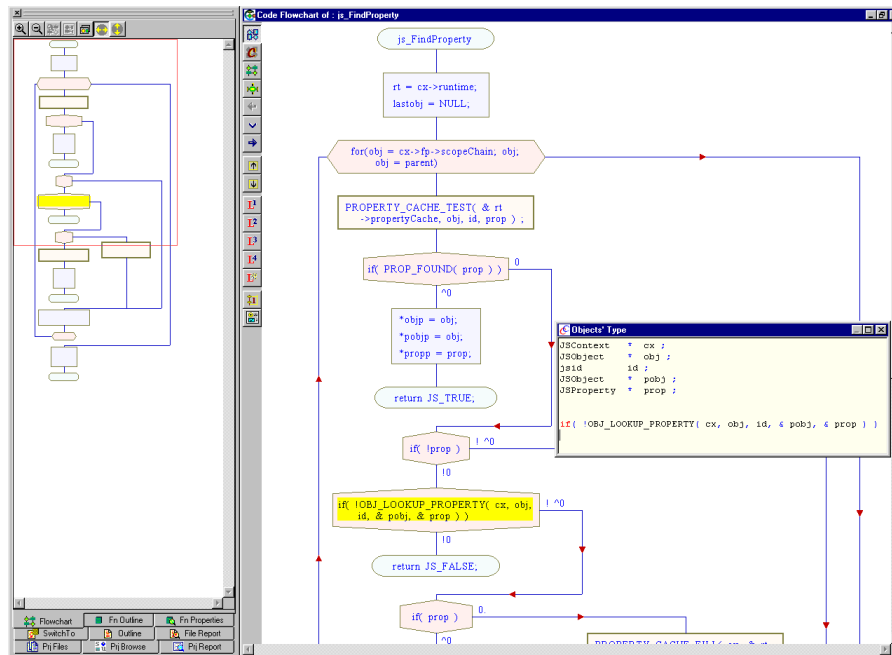


◆ Click on a flowchart symbol to highlight the corresponding code

◆ Click  or  to scroll highlighted code.

## View Object's Type

Also, long statements are easier to read



Because of limited width of symbols, sometimes it is hard to read a lengthy if-expression or a long function-call.

Click in the left half of the symbol.

- The pop-up window displays the code in an easy to read format.
- It also displays the type information of all objects that appear in the symbol.
- In case of a high-level symbol, the pop-up window displays the code covered by that symbol.

A click in the right-half simply selects the symbol.

To export a flowchart as a bitmap file:

Use the "Flowchart" pull-down menu.

Click **Flowchart->Export Flowchart Image->Whole**

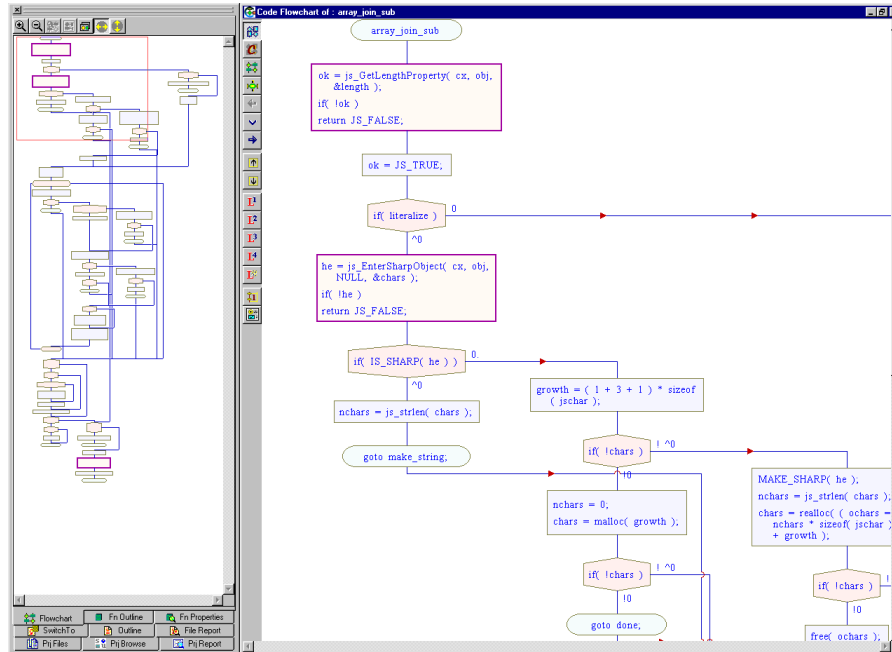
Or drag-and select a part of the flowchart in the detailed view,

Click **Flowchart->Export Flowchart Image -> Selected**

## Part 2: Simplify a Complex Flowchart

1. De-emphasize the goto's
2. Divide and Conquer a Complex Flowchart
  - a. Select Optimal Level of Detail
  - b. View Inner-Code
  - c. Create an If-else Flowchart
  - d. Create a Loop Flowchart
  - e. Expand a High-Level Symbol
3. Zoom-In on a Large switch
4. Export a Flowchart

## A Complex Flowchart



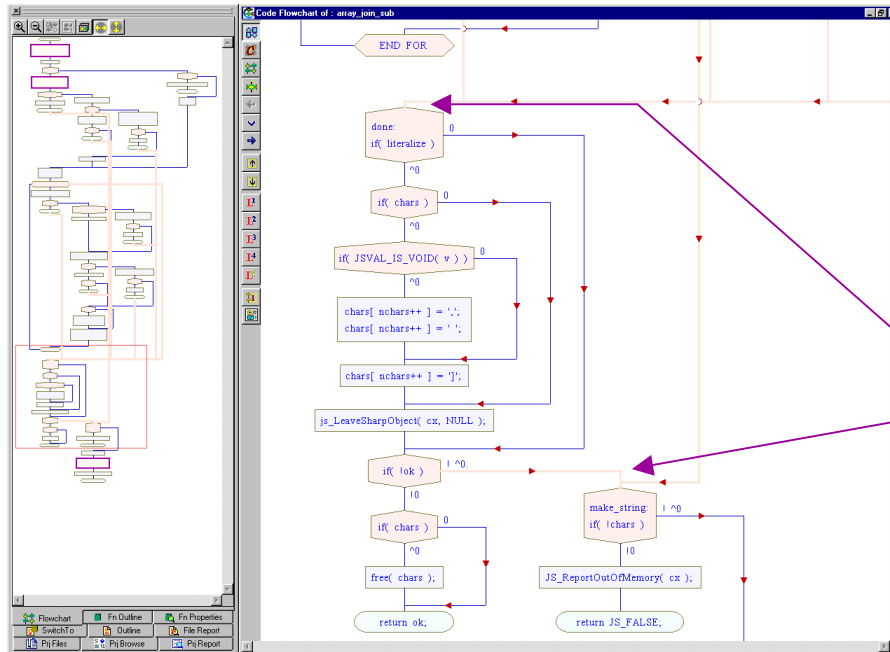
← `array_sub_join()`, a 150-line function.

The function contains many `goto`'s resulting in a complex flowchart.


To simplify a complex flowchart:

- Divide and Conquer
- De-emphasize the `goto`'s

## De-emphasize the goto's



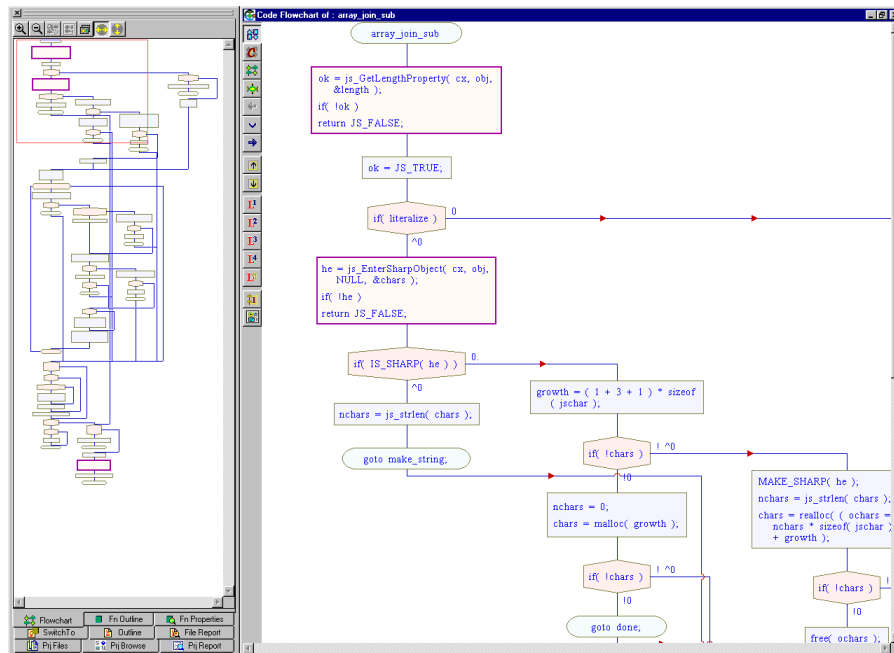
To de-emphasize a connection:

- Click the  button in the toolbar.  
(To change connection-highlighting color)  
Click Highlight Color 2
- Now click on a symbol's input that is the target of a goto.  
(or click on any connection).  
(Press <Ctrl> for additive operation.)

- ◆ With the goto's de-emphasized, it's easy to see the structured parts  
– if-else, Loops etc.  
and it is easy to see the goto's.

## Divide and Conquer a Complex Flowchart

## Select Optimal Level of Detail



Again, we start with the initial flowchart.

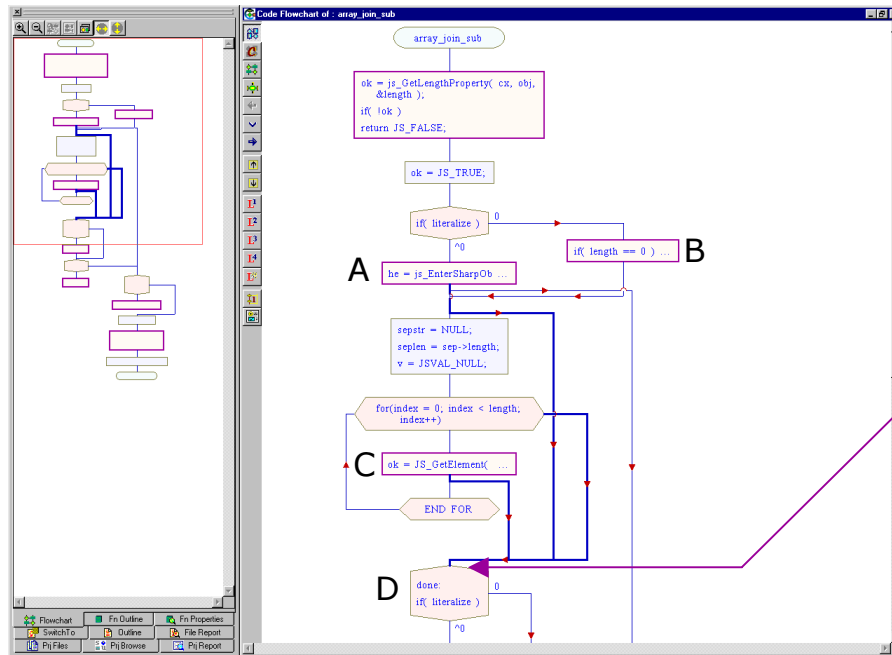
Whenever the condensed view looks crowded, try Level 1 flowchart:

- Click the **L<sup>1</sup>** button in the toolbar.
- If Level 1 flowchart looks too simple, try **L<sup>2</sup>** or **L<sup>3</sup>** so that the flowchart is not too simple nor complex.

Now you have a top-level flowchart that is manageable (next page).



## Level 1 Flowchart



You can read this Level-1 flowchart easily.

To track the connections that are cutting across other connections,

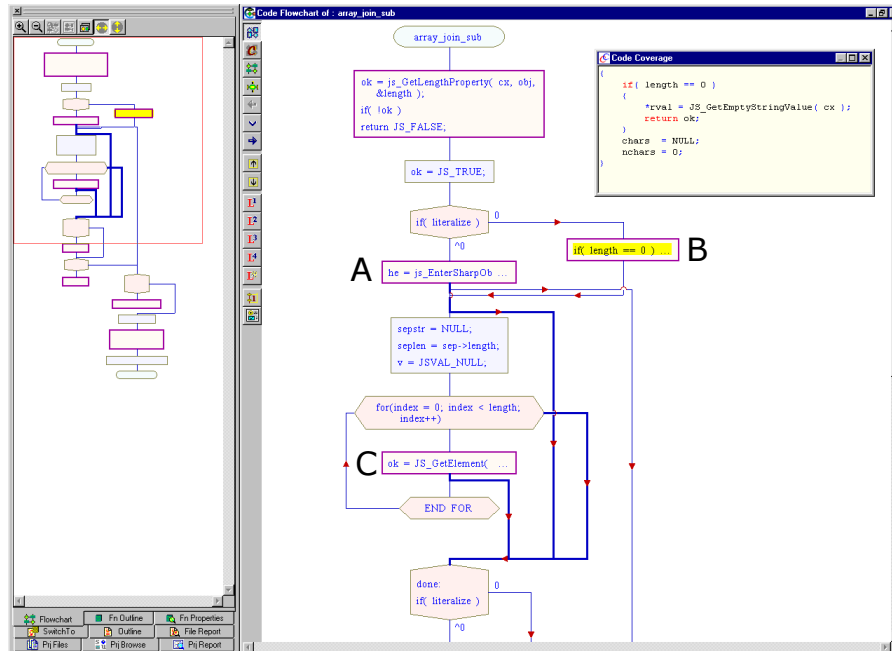
- Click at the input of symbol **D** to highlight its incoming connections.
- Click at the output of symbol **A** to highlight its outgoing connections.
- or
- Click at the output of symbol **B** to highlight its outgoing connection.

**A, B** and **C** are high-level symbols.

A high-level symbol hides the internal details of a loop, switch etc.

A purple outline  indicates a high-level symbol.

## View Inner-Code



Click in the left-half of **symbol B**.

← The pop-up window shows the code covered by **symbol B**.

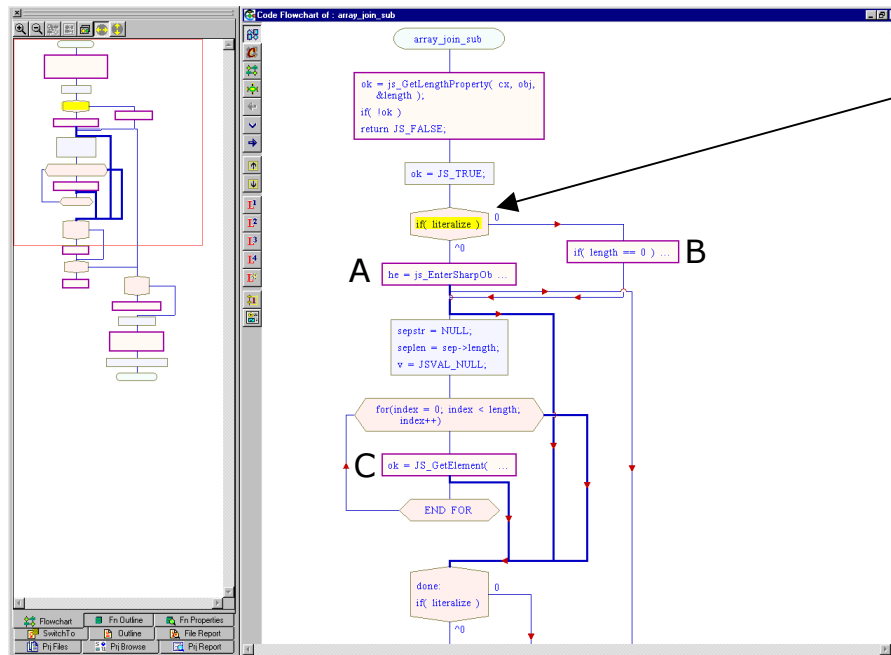
The inner code of **B** is just a few lines.

You can expand **B** (with a double-click) or simply proceed to view the inner code of **A**.

Only **A** and **C** contain sizable code.

- Create flowchart of inner code of **A**.
- then create flowchart of inner code of **C**.

## Create an If-Else Flowchart



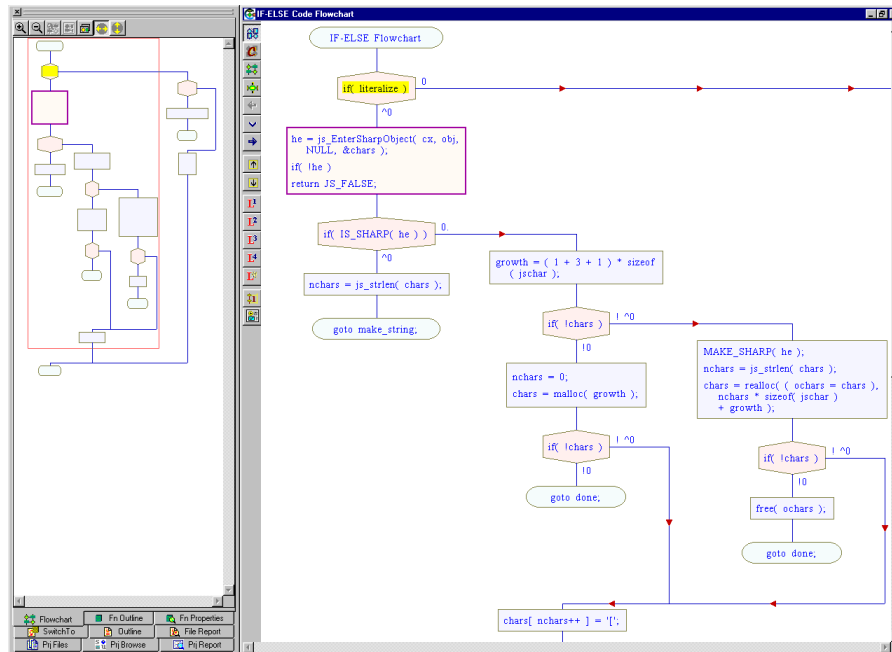
Click in the right-half of the if-symbol to select it;

Click  "Create if flowchart"

In the resulting If-Else flowchart,

- The "YES" part will show the code covered by **A**.
- The "else" part will show the code covered by **B**.

## The Flowchart of Inner Code of **A** and **B**



← In this If-Else flowchart,

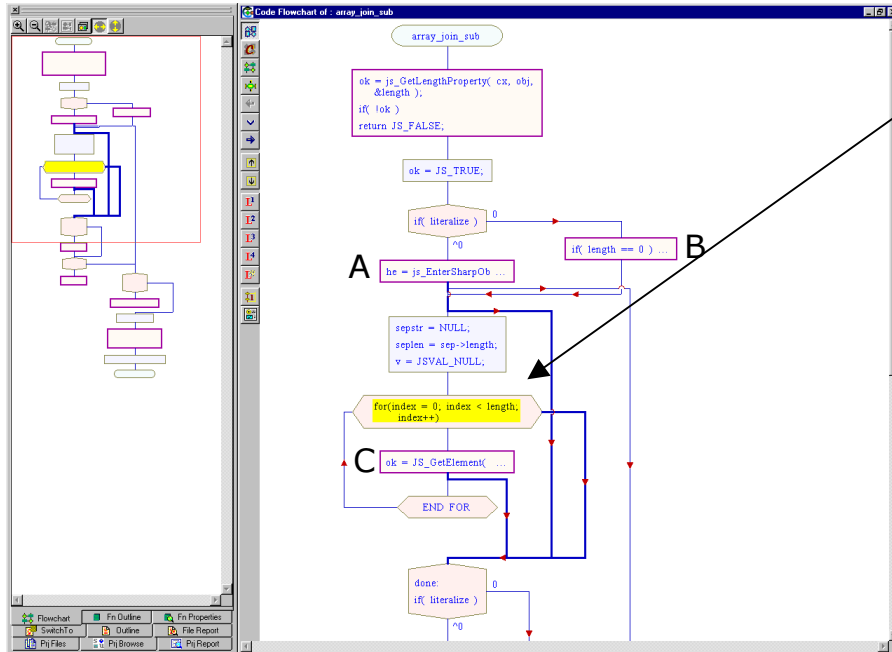
The “YES” part is the code covered by **A**.

The “else” part is the code covered by **B**.

The IF-Else flowchart ends in:

- an **End symbol**: it represents the statement that follows the if-else logic.
- Any **goto's** whose target is outside the If-Else flowchart.

## Create a Loop Flowchart



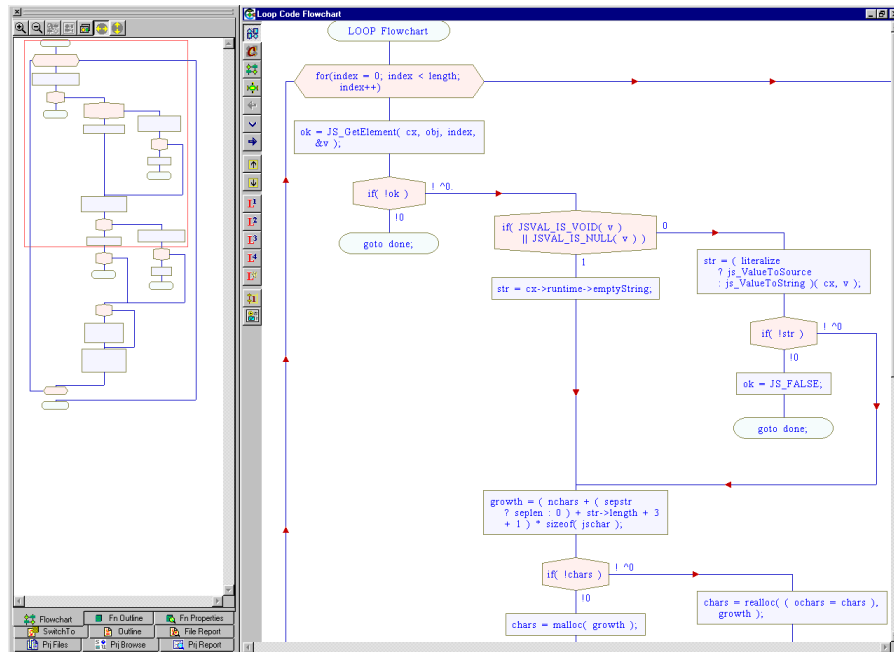
Click in the right-half of the for-symbol to select it.

Click  "Create Loop flowchart"

Similarly, you can create a flowchart for

- a while-loop
- a do-while
- a switch
- a case or
- a compound-statement.

## Flowchart of Inner Code of Symbol C



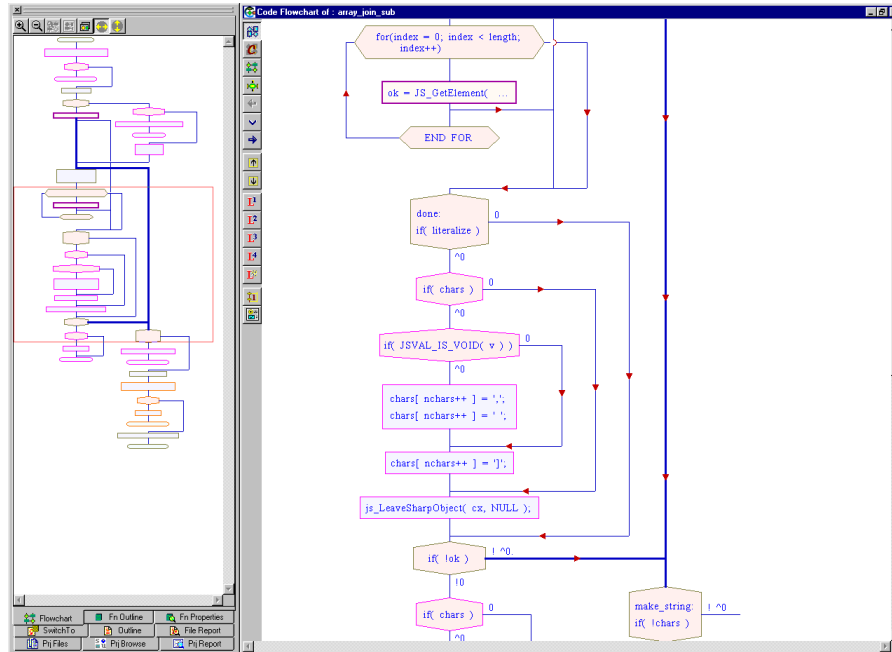
← In this Loop flowchart,

The body of the loop shows the code covered by **C**.

The Loop flowchart ends in:

- an **End** symbol: it represents the statement that follows the loop.
- Any **goto's** whose target is outside the Loop flowchart.

## Expand a High-level Symbol to view inner detail



To expand a high-level symbol:

- right-click on the symbol, then click **Expand Symbol**.
- or **double-click** in the right-half of the symbol.

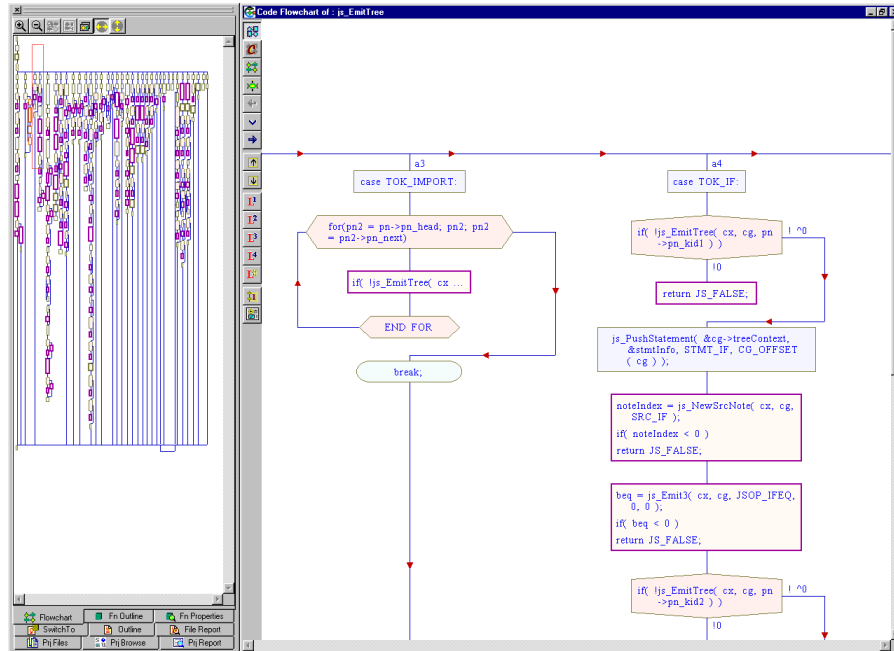
← Here, all high-level symbols **except A and C** have been expanded.

When you collapse a symbol: all consecutive symbols at that indent-level are replaced by a high-level symbol.

An exception - you can collapse a switch by itself.

- To collapse a high-level symbol, right-click on it, then click "Collapse Symbol".
- To collapse an ordinary symbol, double-click in right-half of the symbol.


## Zoom In on a Large switch



← Level 3 flowchart of `js_EmitTree()`,  
a 2000-line function containing a huge  
switch statement.

The condensed-view is too crowded.

You can zoom-in on the condensed view:

- Click Zoom In button  repeatedly.
- or Press the Shift key and drag-and-select the area of interest in the condensed view.

Right-click anywhere in the condensed view, then click Zoom Selection.



## Part 3: Examples

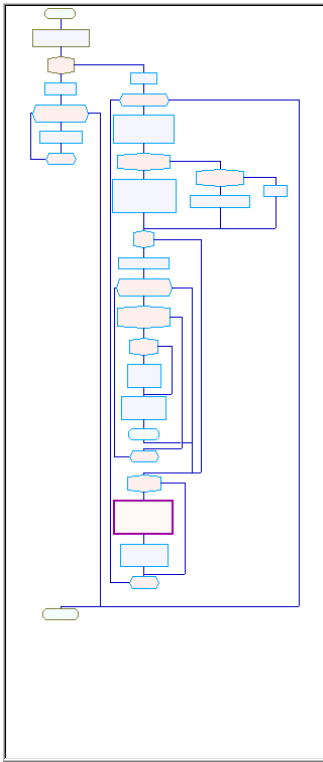
Example-1 **A Moderate-sized Flowchart**


Example-2 **The Function Contains a Switch**

Example-3 **A Very Long Function**

Example-1 **A Moderate-sized Flowchart**

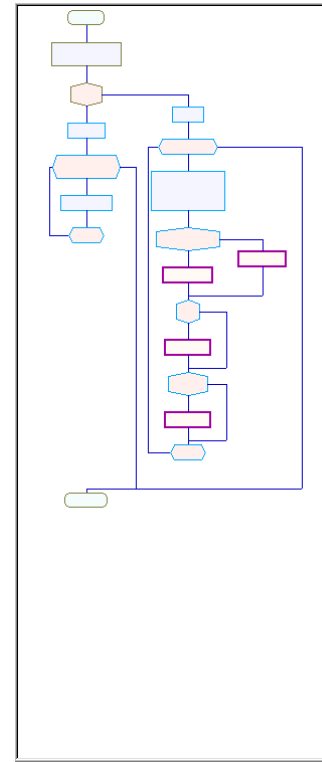
Fig. 1.1

**Initial Optimal-level flowchart**

- ◆ Crystal C creates the initial flowchart as per its **optimal-level-of-detail** algorithm.  
(It corresponds to the **L\*** button in the toolbar)
- ◆ The above flowchart is not too crowded; you can read the flowchart as it is, or you may choose to simplify it. 

**Choose the Level of Detail**

Fig. 1.2

**Level-3 Flowchart**

- ◆ To reduce the amount of detail:
  1. Click the **L<sup>1</sup>** button in the toolbar.
  2. If Level-1 flowchart looks too simple, try **L<sup>2</sup>**, **L<sup>3</sup>** or **L<sup>4</sup>** till the flowchart is not too simple nor complex.  
**L<sup>3</sup>** resulted in the flowchart shown above.



### Example-1 View the Inner code of a High-level Symbol

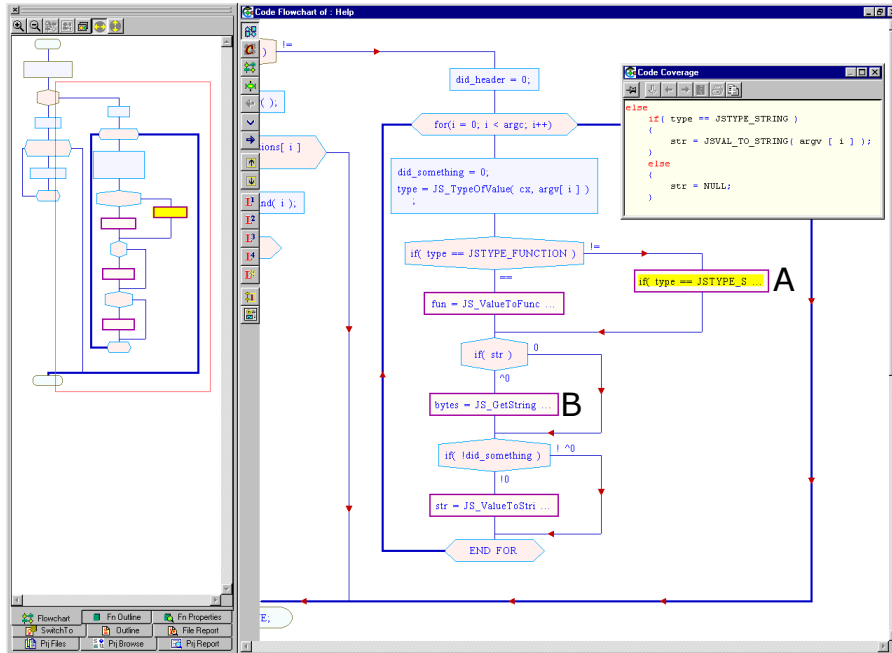


Fig. 1.3

To view the inner detail of symbol **A**:


Click in the left-half of symbol **A**.

← The pop-up window shows the code covered by symbol **A**.

The inner code of A is just a few lines.

- ◆ You can expand **A** (with a double-click) or having seen the code covered by **A**, you can go to next high-level symbol.

Only **B** contains a sizable amount of code.

- A high-level symbol hides the internal details of a loop, switch etc.
- A purple outline  indicates a high-level symbol.

## Example-1 Go through the simplified Flowchart

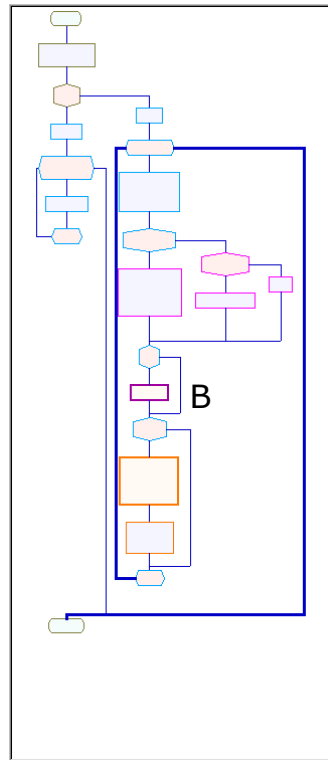


Fig. 1.4

- ◆ To bracket the for-loop: press the ALT key and click the for-symbol in the detailed view.
- ◆ To expand high-level symbols: double-click on high-level symbols in the detailed view. (except B since it contains a non-trivial amount of code)

Now you can go through the above flowchart and then view the flowchart of **B** separately.

## Create an if-else flowchart

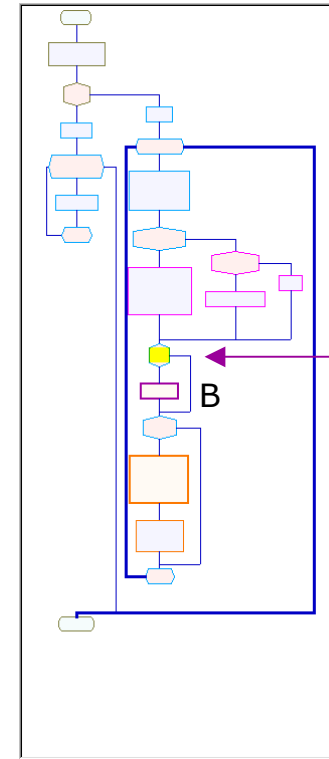



Fig. 1.5

Click the if-symbol

- ◆ To view the flowchart of **B** separately:

1. Click to select the if-symbol under which **B** is nested.
2. Click  to create if-flowchart.



Example-1 Go through the if-else Flowchart (contains B)

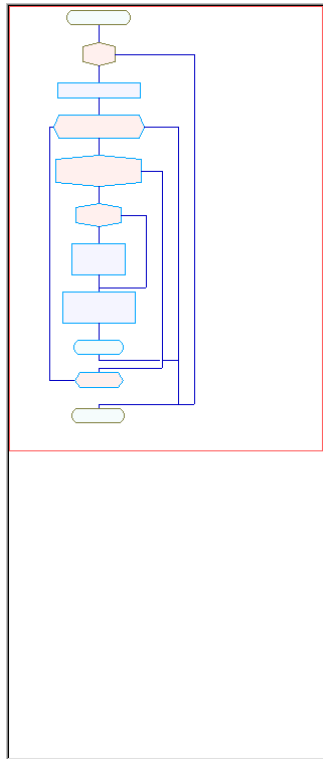
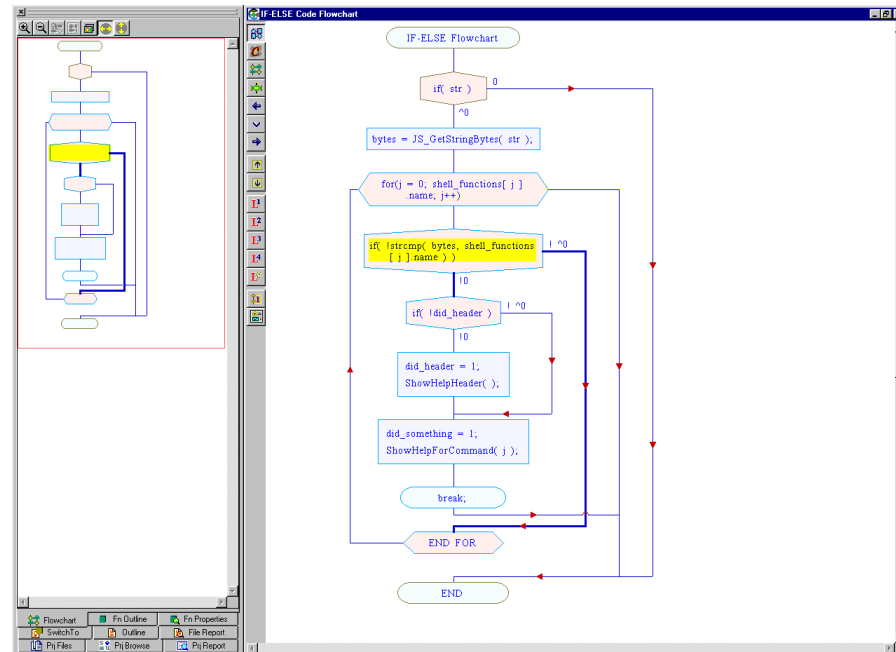


Fig. 1.6

Condensed view of the if-else flowchart


Fig. 1.7



- ◆ Bracket the if as shown above.

It provides a visual marker in an otherwise monotonous flowchart.

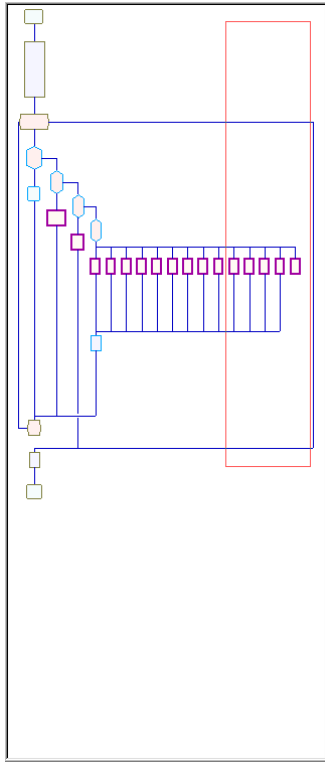
Now you can easily read the above if-else flowchart.

- ◆ To go back to the parent flowchart, click the  button.

(Intentionally Blank)

## Example-2 The Function contains a Switch

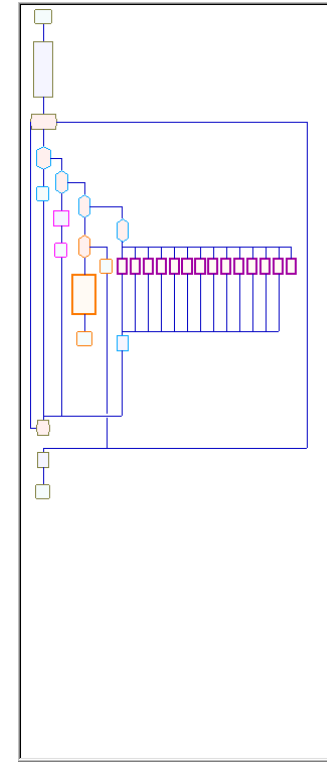
Fig. 2.1



**Initial Optimal-level flowchart**

- ◆ The above flowchart is fairly simple. It contains a switch statement; each case is represented by a high-level symbol.

Fig. 2.2



- ◆ Expand the two high-level symbols: (double-click on high-level symbols in the detailed view.)

Example-2 Create a switch flowchart

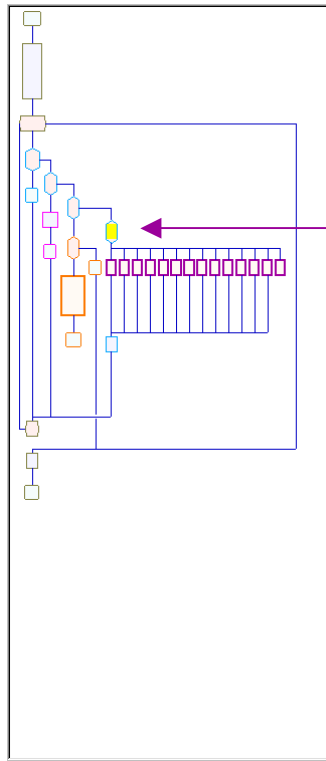


Fig. 2.3

Click the switch-symbol

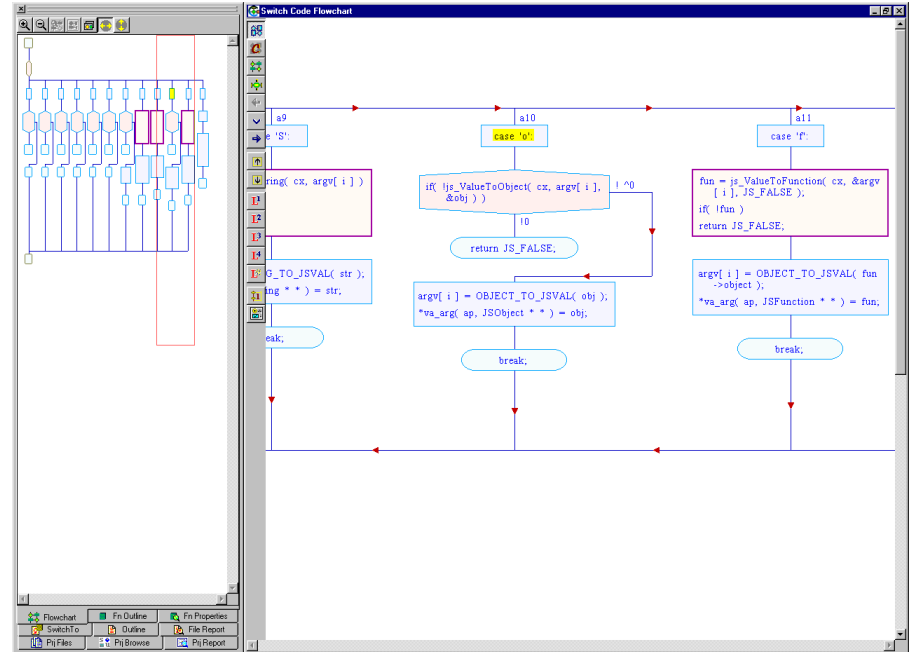



Fig. 2.4

◆ To create switch statement's flowchart:

1. Click to select the switch-symbol.

2. Click  to create switch-flowchart.

◆ Use the cursor keys on your keyboard to move about in the detailed flowchart.

Click on a case-symbol in the detailed view. It will be highlighted and help you as a visual marker.

In this way, you can go through the flowchart.

◆ To go back to the parent flowchart,

click the  button.



## Example-2 Back to the Parent Flowchart

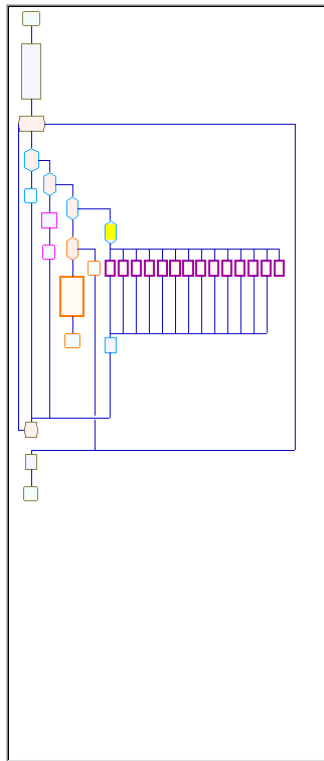


Fig. 2.5

- ◆ Now **collapse** the switch statement:
- 1. **Double-click on the switch-symbol in the detailed view.**
  - To collapse an ordinary symbol, double click on the symbol in the detailed view.
  - To collapse a high-level symbol, right-click on it; then click <Collapse Symbol> in pop-up menu.

## The Whole Function (with the switch collapsed)

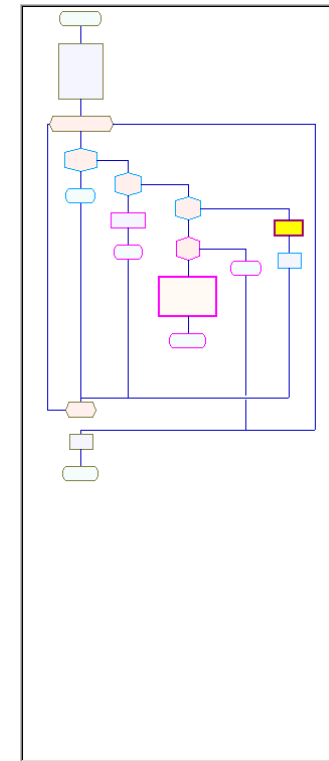


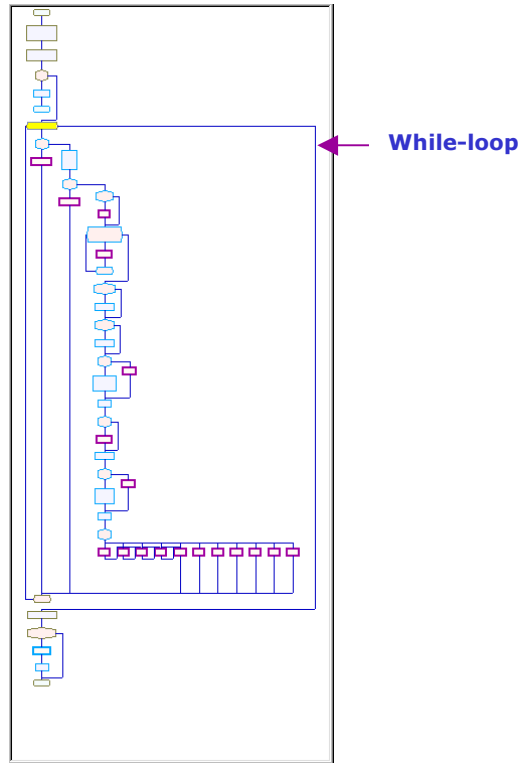
Fig. 2.6

Above is the whole function. (The switch is collapsed.)  
(You saw the details of the switch earlier.)


- ◆ **About collapsing:**
  - When you collapse a symbol, other symbols at its level get collapsed too.
  - However, when you collapse a switch-symbol, only the switch is collapsed.

Example-3 **A Very Long Function**

Fig. 3.1



**Initial Optimal-level flowchart**

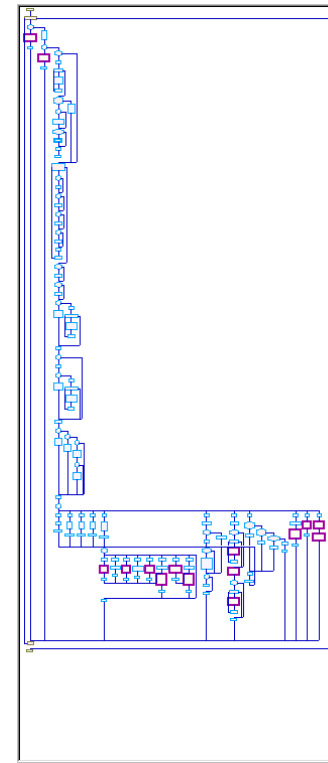
1. Above is the top-level view of a 400 line function.
2. Click on the while-symbol to select it.
3. Click  to create loop-flowchart.

Now you have the flowchart of the while-loop.



**while-loop's Flowchart**

Fig. 3.2



**Initial Optimal-level flowchart of the while-loop**

The initial flowchart looks crowded.

4. Click **L1**, **L2**, **L3** or **L4** till the flowchart is not too simple nor complex.

Example-3 **The while-loop's  
Level-2 Flowchart**

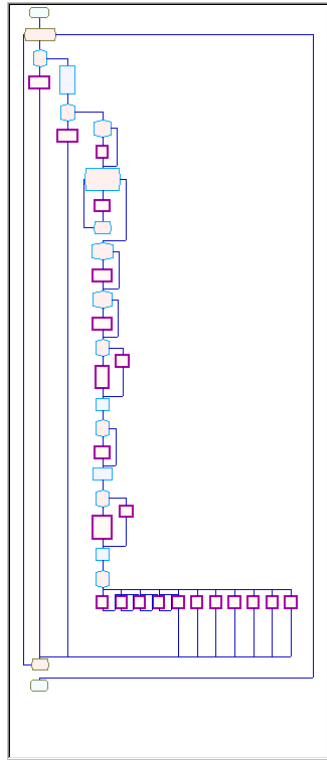


Fig. 3.3

**Condensed View**

In the detailed view:

5. click in the left half of each high-level symbol to see its inner code.
6. In case the inner code is very short, double-click the high-level symbol to expand it.

**After expanding  
some high-level symbols**

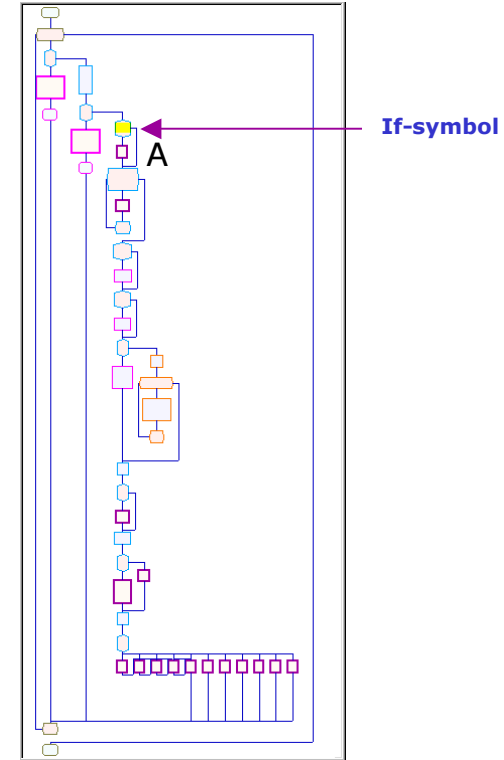



Fig. 3.4

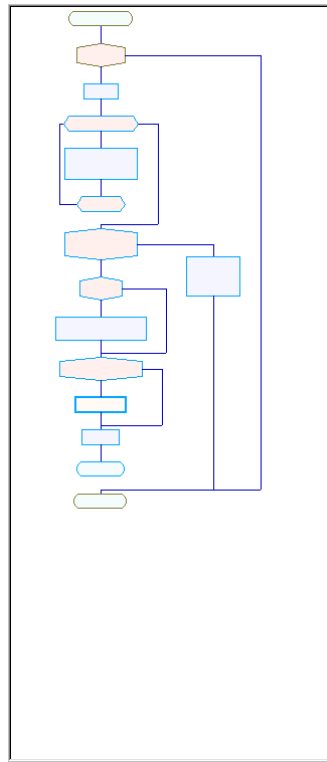
The high-level symbols whose inner-code was very short have been expanded.

To view the flowchart of high-level symbol A:


7. Click to select the if-symbol under which A is nested.
8. Click the  icon in Flowchart toolbar.

Example-3 **If-else Flowchart**  
(shows the details of **A**)

Fig. 3.5



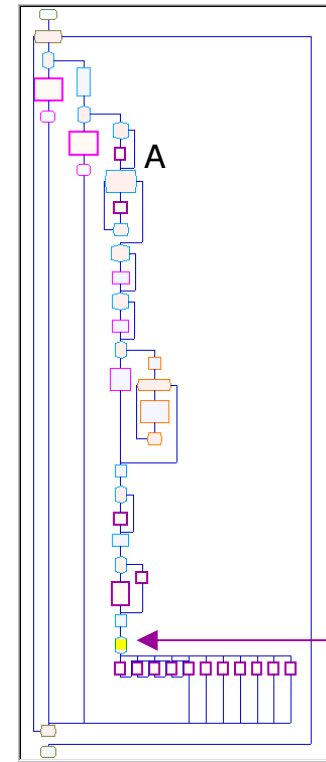
Condensed View

9. Go through the detailed view to see the details of **A**.
10. Click the  button to go back to while-loop flowchart.




**Back to**  
**while-loop's Flowchart**

Fig. 3.6



switch-symbol

Now view the flowchart of the switch:

11. Click to select the switch-symbol.
12. Click  to create switch-flowchart.

Example-3 **switch Flowchart**

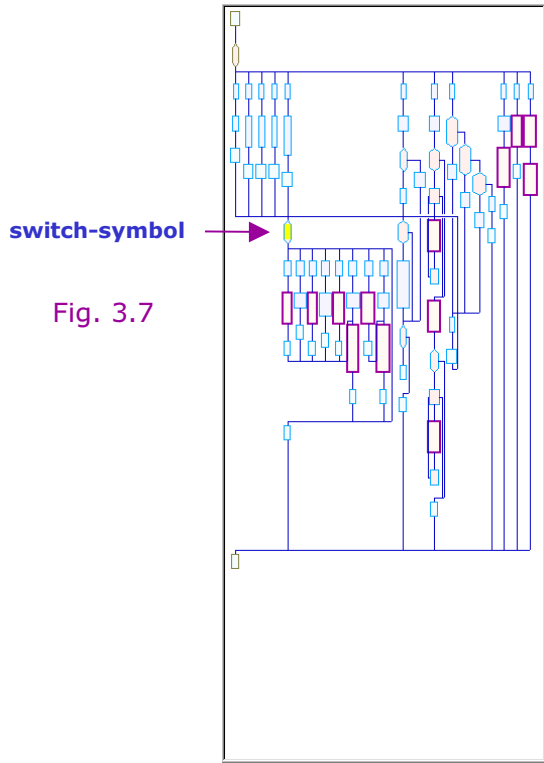



Fig. 3.7

**Condensed View**

The switch contains a nested switch.

First create the flowchart of inner switch:

13. Click to select the switch-symbol.

14. Click  to create switch-flowchart.

**The nested switch**

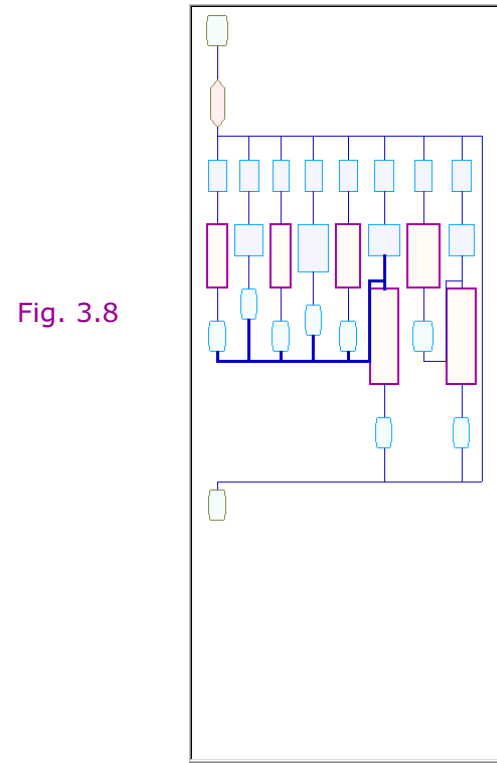


Fig. 3.8

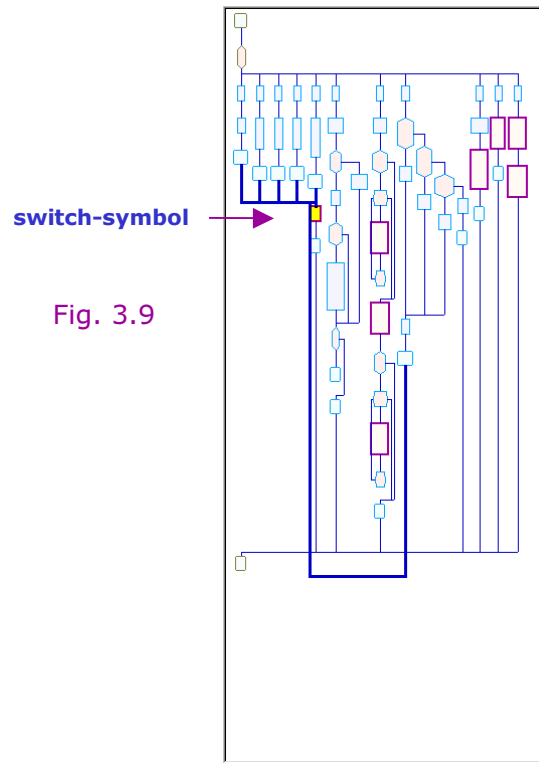
**Condensed View**

The condensed view is simple;  
you can easily go through the detailed view.

and then

15. Click the  button to go back to outer switch flowchart.

## Back to the Outer Switch



Condensed View


- ◆ Now collapse the inner switch:

In the detailed view.:

16. Double-click on the switch-symbol.
17. Click on the input of the high-level symbol to highlight all paths that go to it.

You can easily go through the detailed view.

### Notes:

- To collapse an ordinary symbol, double click on the symbol in the detailed view.
- To collapse a high-level symbol, right-click on it; then click <Collapse Symbol> in pop-up menu. (a double-click on a high-level symbol expands it)
- When you collapse a symbol, other symbols at its level get collapsed too.
- However, when you collapse a switch-symbol, only the switch is collapsed.
- ◆ If you wish to collapse a switch and also wish to create the flowchart for that switch,
  - First create the flowchart for the switch;
  - Go back to the parent flowchart by clicking the  button; then collapse the switch.
  - You may select a case symbol and create the flowchart of that case.

## To Read Simple Flowcharts

When you are going through a simple flowchart  
( or a simplified form of a complex flowchart),

use the following operations:

- ◆ Bracket a loop or bracket an if-else segment of the flowchart.
- ◆ Highlight all paths that can reach a given point in the flowchart.  
Highlight a connection so that it stands out from other neighboring connections.
- ◆ View the type information of all objects that appear in a given symbol.
- ◆ Get a side-by-side view of the flowchart and corresponding code.

(Please refer to [Part 1: Simple Flowcharts.](#))

## To Simplify a Complex Flowchart

1. Change the level of detail:

Try L1, L2, L3 or L4 from the toolbar till the flowchart is not too simple nor complex.

2. View the inner code of high-level symbols:

If the inner code is just three or four lines,  
you may wish to expand the symbol or leave it as it is.

3. If the inner code contains a substantial number lines,

Create the flowchart of the "if" or the loop that encloses the high-level symbol.

4. If the flowchart contains a switch statement,

Create a flowchart of the switch, then go back to parent flowchart, collapse the switch.