## MSVC IDE Debugger List Tracing

### A05. List Debug 1

#### Table of Contents

- Watch Window
- Watching Lists
- Watching Linked-List Variables
- Dereferencing Invalid Addresses





#### Using the Watch Window to View List Structures

- Once the program is successfully compiled and ready to execute:
- Set breakpoints at the locations(s) in the code where you wish to view the list's contents. Position the cursor on a line of code, right click, and select **Insert/Remove Breakpoint**.
- Select Build -> Debug -> Go from the menu or hit F5 to begin debugging.

<u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp					
<u>C</u> ompile listview.cpp <u>B</u> uild listview.exe <u>R</u> ebuild All Batch Build	Ctrl+F7 F7	- 	14 14 14 12 14 14		
Stop <u>B</u> uild Update <u>A</u> ll Dependencies	Ctrl+Break.	se of the			
<u>D</u> ebug Execute listview.exe	Ctrl+F5	Go Step Into	F5 F11		
<u>S</u> ettings C <u>o</u> nfigurations	Alt+F7	Run to Cursor	Ctrl+F10		
Subprojects Set Default Configuration			<u>V</u> iew Insert	<u>D</u> ebug <u>T</u> o	ools <u>W</u> indow
		]	Class <u>W</u> izaro	d	Ctrl+W/
			Resource S Resource I <u>r</u>	iymbols acludes	
The program will stop at the first			F <u>u</u> ll Screen		
breakpoint. If the Watch Window is not visible, select <b>View -&gt; Watch</b> from the menu.		<u>I</u> oolbars			
		InfoViewer Query Results InfoViewer History List			
		Project Wor	rkspace	Alt+0	
			IntoViewer	l opic	Alt+1 Alt+2
		Watch		Alt+3	
			⊻ariables	13	Alt+4
			Disassemble	U	Alt+8

#### Viewing Linked List Contents (continued)

- Double-click the structure variable name in the Source window to highlight it, then drag and drop it into the Watch
  - Window.

Name	Value	
+ listhead	0x00000000	
h	<b>*</b>	
		_
Watch1 (Wat	tch2 $\lambda$ Watch3 $\lambda$ Watch	4/

There will be a plus (+) sign to the left of the variable name in the *name field* of the Watch Window. Click on the plus (+) to "expand" the variable. This will in effect display the "contents" of the structure with each subsequent expansion of the variable. A minus (-) sign indicates that the variable is already fully expanded.

Name	Value 🔺		
📮 listhead	0x00760154		
- val	0		
- nextnode	0x00760184		
- val	1		
L	Oxededed		
l K			
₩atch1 (Watch2 \Watch3 \Watch4 /			

## Watching Linked-List Variables

#### Sample Program

```
/* Sample program to demonstrate use of the
    variables window. */
#include <iostream.h>
typedef struct node {
    int val;
    node* nextnode;
};
int i;
node* listhead, * currnode;
void main (void)
                              create a list of size 3
ł
    listhead = new node;
                              and set the last node
    listhead -> val = 0;
                              pointer to NULL
    currnode = listhead;
    for (i=1; i<=2; i++)</pre>
         currnode - > nextnode = new node;
         currnode = currnode->nextnode;
         currnode->val = i;
     ł
    currnode->nextnode = NULL;
    for (currnode = listhead; currnode != NULL;
                  currnode=currnode->nextnode)
         cout << currnode->val << " ";
    return;
}
```

# A05. List Debug

5

## Dereferencing Invalid Addresses

 The tail of the list, when set to NULL will appear in the watch window as 0x00000000. Any values at this point in the list will be inaccessible.



 If the last node is NOT set to NULL, the values will still be accessible but will most likely produce a run-time error or undesired program behavior, even in the debugger.

currnode = currnode->nextnode;				
currnode->val = 1;				
} // last node not set to NULI				
for (currnode = listhead;currnode != NULL;				
currnode=currnode->nextnode)				
cout << currnode->val << " ";				
MICI	Microsoft Developer Studio			
Name				
🕞 listhead 🛛 🚺	Unhandled exception in listview.exe: 0xC0000005: Access Violation.			
- val	· ·			
- nextnode				
- val	OK			
- nextnod				
val	2			
- next node				
	CV0020 Empone our connet be evaluate			
- vai	CANDUSU. Error: expression cannot be evaluate			
nextnode	CXX0030: Error: "expression cannot be evaluate			