

Exhibit CC-2

\\BATEABC01929\ABC\Source\CPlusCopy.cpp

```
1  /*
2  *    CPlusCopy.cpp
3  *    C++ Copy Routine
4  *    Copyright 2003, Good Guys, Inc. All rights reserved.
5  */
6
7  #include <CPlusLib.h>
8
9  /*
10 *    copy - copy 'length' number of bytes from 'from
11 *    ' to 'to'
12 *    If the destination or length is zero, don't copy anything.
13 *    Return the destination.
14 */
15
16 void *copy(char *to, char *from, length_t length)
17 {
18     char *p;
19
20     if (to && length) {
21         p = to;
22         do {
23             *p = *from;
24             ++p;
25             ++from;
26             --length;
27         } while (length);
28     }
29
30     return(to);
31 }
32
33 /*
34 *    constructObjects - set up array of objects
35 *
36 *    Call the constructor for each of 'numberOfObjects'
37 *    in an array
38 *    of objects that are 'sizeofObjects' long.
```

\\BATEXYZ45724\XYZ\Lib\objectUtils.cpp

```
1  /*
2  *    objectUtils.cpp
3  *    Utility Routines
4  *    Copyright 2003, Bad Guys, Inc. All rights reserved.
5  */
6
7  #include <CPlusLib.h>
8
9
10 /*
11 *    callConstructors - construct array of objects
12 *
13 *    Call the constructor for each object in the array
14 */
```

Exhibit CC-2

\BATEABC01929\ABC\Source\CPlusCopy.cpp

```
38  */
39
40  void *constructObjects(void *ptr, ConstructorDestruct
▶ or constructor, length t sizeofObjects, length t numb
▶ erOfObjects)
41  {
42      char *p;
43
44      if ((p = (char *) ptr) != 0) {
45          ((length t *) p)[0] = sizeofObjects;
46          ((length t *) p)[1] = numberOfObjects;
47          p += 2 * sizeof(length t);
48          if (constructor) {
49              for (; numberOfObjects--> p += sizeofObje
▶ cts)
50                  CTORCALL COMPLETE(constructor, p);
51          }
52      }
53
54      return(ptr);
55  }
```

\BATEXYZ45724\XYZ\Lib\objectUtils.cpp

```
14  */
15
16  void *callConstructors(void *ptr, ConstructorDestruct
▶ or constructor, length t objectSize, length t howMany
▶ )
17  {
18      char *p;
19
20      if ((p = (char *) ptr) != 0)
21          {
22              ((length t *) p)[0] = objectSize;
23              ((length t *) p)[1] = howMany;
24              p += 2 * sizeof(length t);
25              if (constructor)
26                  {
27                  for (; howMany--> p += objectSize)
28                      CTORCALL COMPLETE(constructor, p);
29              }
30          }
31
32      return(ptr);
33  }
34
35  /*
36  *    duplicate - duplicate 'length' number of bytes
▶ source 'source' dest 'dest'
37  *
38  *    If the destination or length is zero, don't dup
▶ licate anything.
39  *    Return the destination.
40  */
41
42  void *duplicate(char *dest, char *source, length t le
▶ ngth)
43  {
44      char *p;
45
46      if (dest && length)
47          {
48              p = dest;
```

```
49         do
50     {
51         *p = *source;
52         ++p;
53         ++source;
54         --length;
55     } while (length);
56 }
57
58     return(dest);
59 }
```

=====
Lines of Code:

Total	59
Literally Identical/Translated	33 = 55.9%
Functionally Identical	13 = 22.0%
Filtered	1 = 1.7%

The following translation equivalents were found and used in highlighting this file:

- Good=Bad
- constructObjects=callConstructors
- to=dest
- copy=duplicate
- numberOfObjects=howMany
- sizeOfObjects=objectSize
- CPlusCopy=objectUtils
- from=source

Note: During formatting tabs are converted to four spaces and all lines longer than 53 characters are wrapped. All wrapped lines are denoted with a ► character at the beginning of the line; however, highlighting is based on the full line prior to formatting.