**Handling Escaped Values for the Interpreter**

For the interpreter, you have to convert escaped values to the proper char value. Expected escaped values:

\" " double quote

\' ' apostrophe

\\ \ back slash

\n 0x0A line feed

\t 0x09 tab

\a 0x07 alarm (bell)

Strongly consider copying characters from the literal into a char array. The number of characters in that char array will be less than the characters in the original HavaBol string if that string contained escaped characters. For example, the literal "\tTX\tTexas\n" contains three escaped characters. You must replace those three (in this example) with single byte hex values.

Java code: retCharM[iRet] = 0x09; // tab character

To copy a char array into a String, use:

tokenStr = String.valueOf(retCharM // char array

,0 // beginning subscript in the char array

,iRet); // number of characters in the char array to copy

You might want a function to print OPERAND STRING literals, see below.

Some sample output for Program #2:

9 String weird;

SEPARATOR - ;

CONTROL DECLARE String

OPERAND IDENTFIER weird

10 weird = "\tTX\tTexas\n";

SEPARATOR - ;

OPERAND IDENTFIER weird

OPERATOR - =

OPERAND STRING . TX. Texas.

09 09 0A

Example hexPrint funtion in Java:

/\*\*

\* Prints a string that may contain non-printable characters as two lines.

\* <p>

\* On the first line, it prints printable characters by simply

\* printing the character. For non-printable characters

\* in the string, it prints ". ".

\* <p>

\* The second line prints a two character hex value for the non printable

\* characters in the string line. For the printable characters, it prints

\* a space.

\* <p>

\* It is sometimes necessary to print the first line on the end of

\* an existing line of output. This would make it difficult to properly

\* align the second line of output. The indent parameter is for indenting

\* the second line.

\* <p><blockquote><pre>

\* Example for the string "\tTX\tTexas\n"

\* . TX. Texas.

\* 09 09 0A

\* </pre></blockquote><p>

\* @param indent the number of spaces to indent the second printed line

\* @param str the string to print which may contain non-printable characters

\*/

public void hexPrint(int indent, String str)

{

int len = str.length();

char [] charray = str.toCharArray();

char ch;

// print each character in the string

for (int i = 0; i < len; i++)

{

ch = charray[i];

if (ch > 31 && ch < 127) // ASCII printable characters

System.out.printf("%c", ch);

else

System.out.printf(". ");

}

System.out.printf("\n");

// indent the second line to the number of specified spaces

for (int i = 0; i < indent; i++)

{

System.out.printf(" ");

}

// print the second line. Non-printable characters will be shown

// as their hex value. Printable will simply be a space

for (int i = 0; i < len; i++)

{

ch = charray[i];

// only deal with the printable characters

if (ch > 31 && ch < 127) // ASCII printable characters

System.out.printf(" ", ch);

else

System.out.printf("%02X", (int) ch);

}

System.out.printf("\n");

}