User Interface - Bug #2193

message with view-as alert-box visual errors when using special characters

10/14/2013 08:16 AM - Vadim Gindin

Status:	New	**Start date:**	10/14/2013
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

History

#1 - 10/14/2013 11:18 AM - Vadim Gindin

- File first.png added

- File second.png added

- File third.png added

Here is the problem test.

```
message "1 \n\r2 \n\r3" view-as alert-box.
message "11 \r\n12 \r\n13" view-as alert-box.
message "21 \r22 \r23" view-as alert-box.
```

I've attached 3 screen shots to show sequenced screens (see png's).
When you'll see these images you'll find that there are several errors there:
1) Probably \r as a carriage return character really moves carriage to beginning of new line and ignores alert-box boundaries. 2 and 3 are in the start of the lines.
2) The second alert-box is almost correct except on space before 13 and of course it didn't erase 2 and 3 from the first screen
3) The third screen is the most cryptic. 22 is gone. and 23 overwrites 2 from the first screen.

I suspect that the reason can be in a terminal specific features.

#2 - 10/16/2013 11:59 AM - Vadim Gindin

1. First problem is wrong visual effects that appears when we use message "somestr" view-as alert-box statement. This problem is described above.

2. Have following testing procedure.

```
output to skip_test.
```

```
message "first string " skip
```

```
        "second string " skip
        "third string " skip.
```

From the Progress docs:

```
SKIP [ ( n ) ]
```

```
   Indicates a number (n) of blank lines to insert into the message. The value of n can
   be 0. If you do not specify n, or if n is 0, a new line is started unless the current
   position is already the start of a new line.

   You can only use this option with the VIEW-AS ALERT-BOX option.
```

The problems are:
2a. Progress error. The output string is truncated in terminal during execution of the previous procedure.

```
first string 
```

It seems, that Progress processes skip statement only substituting it with '\n' in a message string. Here is a analogue statement message "first\nsecond\nthird".

2b. If the source message statement with skip is used with view-as alert-box and the analogue statement is also used with view-as alert-box. The only difference in Progress behavior is the different text alignment in the alert box. In the first case the alignment in the alert box is by right-boundary and in the second case - is by left-boundary.

2c. Runtime error. The wrong output in a resulting file skip_test after execution converted Java class.

```
first string  com.goldencode.p2j.ui.SkipEntity@9e7d46 second string
com.goldencode.p2j.ui.SkipEntity@132b038 third string
com.goldencode.p2j.ui.SkipEntity@cf5006
```

3. Having the next test:

```
message "111 \r222 \r333".
```

The Progress outputs the string 333. It needs to find out why it does so.

4. Slashes problem. Having test:

```
output to skip_test.
 message "one \ntwo \nthree".
 message "1 \n\r2 \n\r3".
 message "11 \r\n22 \r\n33".
 message "111 \r222 \r333".
```

The converted Java class contains:

```
message("one \\ntwo \\nthree");
message("1 \\n\\r2 \\n\\r3");
message("11 \\r\\n22 \\r\\n33");
message("111 \\r222 \\r333");
```

Extra slashes was added during conversion. The Java strings are not the same as source Progress strings. The Java won't see the special char \n, that is in the Progress string. Java will see two symbols: escaped slash \ and the n character.

Files

first.png	5.5 KB	10/14/2013	Vadim Gindin
second.png	4.24 KB	10/14/2013	Vadim Gindin
third.png	4.02 KB	10/14/2013	Vadim Gindin