Nirva Application Platform - Bug #70

Truncation of Commands returned String in java interface

07/23/2012 12:17 PM - Pierre Marc

Status:	Closed	**Start date:**	07/23/2012
Priority:	High	**Due date:**	
Assignee:	Pierre Marc	**% Done:**	100%
Category:	Application Platform	**Estimated time:**	0.00 hour
Target version:	4.7.004		
Operating System:	Any	**Tested:**	Yes
Version:			

Description

When doing a OBJECT:??????_GET_VALUES with MODE=|STRING| the resulting string in Java is sometimes truncated.

I have noticed that in the C++ service there is an example which shows using the the 3rd parameter of the command call (a integer pointer) to set a new buffer size if the returned length was greater than the original buffer size. Thereby letting the programmer to create a new buffer to size+1 and recall the command with the new buffer to obtain the full string with no truncation.

Is there any such work around in Java to deal with the string truncation, other than either having to step through each element of the object or using MODE=|VARIABLE| and doing countless calls to VARIBLE:GET?

History

#1 - 07/23/2012 12:18 PM - Pierre Marc

here is demonstration Java procedure that produces the issue:

```
//import com.nirvasoft.nirva.nvcmd;
import com.nirvasoft.nirva.nvsint;
/* PROCEDURES */
class test_truncate
{
 public static int main()
 {
  nvsint ProcCMD;

  // Procedure Class
  ProcCMD = new nvsint();

  if(ProcCMD == null)
  {
    System.out.print("Error getting nvinst class\n");
    return -1;
  }
  // Create a INDSTRINGLIST
  ProcCMD.Command("NV_CMD=|OBJECT:CREATE| TYPE=|INDSTRINGLIST| NAME=|TRUNCATE_EXMP| REPLACE=|YES|");
  // Add a Few Values
  for (int i=0; i<100; i++)
  {
   ProcCMD.Command("NV_CMD=|OBJECT:INDSTRINGLIST_SET_VALUE| NAME=|TRUNCATE_EXMP|
 KEY=|SOME_LONG_KEY_NAME_"+Integer.toString(i)+"_WITH_NO_DESCRIPTION_BLAH_BLAH|
 VALUE=|SOME_SILLY_LONG_VALUE_"+Integer.toString(i)+"_WITH_NO_DESCRIPTION_BLAH_BLAH|");
  }
  ProcCMD.Command("NV_CMD=|OBJECT:INDSTRINGLIST_GET_VALUES| MODE=|STRING| NAME=|TRUNCATE_EXMP|");
  System.out.println(ProcCMD.NvResult);

  return 0;
 }
}
```

The output returned to the console ends with "=SOME_SILLY_VALUE_9_WITH_NO_DESCRIPTION_BLAH_BLA" and interestingly its only the last character that is missing! the whole output this generates is only 10678 chars long, so I was wondering if your using malloc if maybe the calculation to extend the field hasn't taken into account the string null terminator or something?

#2 - 07/23/2012 12:19 PM - Pierre Marc

- Status changed from New to In Progress

This is a bug that occurs from 4.7.002 version. It will be corrected in v 4.7.004.

#3 - 07/25/2012 06:01 PM - Pierre Marc

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

- Tested changed from No to Yes

#4 - 07/26/2012 10:49 AM - Pierre Marc

- Status changed from Resolved to Closed