

Runtime Infrastructure - Bug #5699

broken logfile name on Windows appserver clients (and probably other cases on Windows)

09/30/2021 11:23 AM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Galya B	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			
Related issues:			
Related to Runtime Infrastructure - Bug #5701: logfile names for -O outputToF...		Closed	
Related to Runtime Infrastructure - Bug #5703: rationalize, standardize and s...		Closed	
Related to Runtime Infrastructure - Feature #7147: Make FWD log outputs consi...		New	

History

#1 - 09/30/2021 11:50 AM - Greg Shah

Spawned appserver clients (and batch processes?) can be configured in the server's directory with a log filename for redirection of stdout.

```
<node class="container" name="ffappsrvprocess">
  <node class="container" name="clientConfig">
    <node class="string" name="outputToFile">
      <node-attribute name="value" value="client_%appserver%_%pid%_%timestamp%_%userid%.log"/>
    </node>
  </node>
</node>
```

If this is present, it is processed by `ProcessBuilderParameters.getOutputToFile()` on the server before the launch of the spawner. In the above example, the spec is:

```
client_%appserver%_%pid%_%timestamp%_%userid%.log
```

3 of the placeholders are processed in `ProcessBuilderParameters.getOutputToFile()`:

`%appserver%` - comes from the appserver entry in the process account, if it is there
`%timestamp%` - `System.currentTimeMillis()`, which seems to be there
`%userid%` - the FWD process account name, which seems to be there as `ffappsrvprocess`

The process account:

```
<node class="process" name="ffappsrvprocess">
  <node-attribute name="appserver" value="ffappsrv"/>
  <node-attribute name="enabled" value="TRUE"/>
  <node-attribute name="description" value="Process for ffappsrv AppServer."/>
  <node-attribute name="alias" value="ffappsrvprocess"/>
  <node-attribute name="master" value="FALSE"/>
  <node-attribute name="server" value="FALSE"/>
</node>
```

So the appserver name should appear as `ffappsrv`.

But the resulting log names (when the clients are spawned on Windows) are in this form:

```
client_f11616_1631901774729_ffappsrvprocess.log
```

The `%pid%` placeholder is swapped out in the spawner. It looks to me like a portion of the spec (`%appserver%_%pid%`) converts into `f11616` when

we would have expected ffappsrv_11616.

I extracted the Linux version of the spawner's replacement code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>

#define MAX_FILENAME 4096

char* replace_str(char* string, const char* placeholder, char* value)
{
    static char buffer[MAX_FILENAME];
    int len;
    char *p;

    p = strstr(string, placeholder);

    if (!p)
    {
        // placeholder not found
        return string;
    }

    len = p - string;
    strncpy(buffer, string, len);
    buffer[len] = 0;

    sprintf(buffer + len, "%s%s", value, p + strlen(placeholder));

    return buffer;
}

int main(int argc, char *argv[])
{
    static char buffer[MAX_FILENAME];

    char* string = "client_ffappsrv_%pid%_1631901774729_ffappsrvprocess.log";
    char* placeholder = "%pid%";

    char value[32];

    /* convert pid to string */
    sprintf(value, "%u", getpid());

    char* filename = replace_str(string, "%pid%", value);

    printf("%s\n", filename);
}
```

When I compile and run this on Linux, the output is OK (client_ffappsrv_148443_1631901774729_ffappsrvprocess.log). I don't have a WIN32 dev environment setup at this time. My best guess here is that the `replace_str()` function in `winspawn.c` has some logic problems. This is probably related to the usage of the WIN32 wide character support in that code.

#3 - 09/30/2021 12:35 PM - Greg Shah

- Related to Bug #5701: logfile names for -O outputToFile use the spawner pid instead of the Java (FWD client process) pid added

#4 - 10/04/2021 07:09 AM - Greg Shah

- Related to Bug #5703: rationalize, standardize and simplify the client-side log file name configuration added

#5 - 11/29/2022 08:36 AM - Greg Shah

- Assignee set to Galya B

#6 - 12/08/2022 11:42 AM - Galya B

- Status changed from New to WIP

I was able to reproduce it. The problem is in `replace_str` function. I've added to the code a few comments to show values of vars runtime, when the functions is executed with the args string = `client_ffappsrv_%pid%_1631901774729_ffappsrvprocess.log`, placeholder = `%pid%`, value = `11616`:

```
LPWSTR replace_str(LPWSTR string, LPCWSTR placeholder, LPWSTR value)
{
    static WCHAR buffer[MAX_SIZE];
    WCHAR *p;
    int len;

    p = wcsstr(string, placeholder);          /////// %pid%_1631901774729_ffappsrvprocess.lo
g
    if (!p)
    {
        // placeholder not found
        return string;
    }

    len = (p - string) / sizeof(WCHAR);      /////// 8
    wcsncpy(buffer, string, len);
    buffer[len] = 0;                          /////// client_f

    swprintf(buffer + len, sizeof(buffer), L"%ls%ls", value, p + wcslen(placeholder)); /////// client_f11616_163
1901774729_ffappsrvprocess.log

    return buffer;
}
```

The calculation of `len` seems to be giving wrong results, which cuts the string after the 8 character before concatenating it with the pid and the rest of the string.

I'm working on revamp of that function to make it easier to understand and maintain.

#7 - 12/08/2022 04:23 PM - Greg Shah

Is string not a pointer to WCHAR? If the contents of string are not wide characters, then that explains the result. For the input string `client_ffappsvr_%pid%_1631901774729_ffappsvrprocess.log`, the placeholder is found 16 characters in (after `client_ffappsvr_`). The `sizeof(WCHAR)` is probably twice as large as the characters in the string, which leads to the `len == 8` result.

#8 - 12/09/2022 05:20 AM - Galya B

- Status changed from WIP to Review

Greg Shah wrote:

Is string not a pointer to WCHAR? If the contents of string are not wide characters, then that explains the result. For the input string `client_ffappsvr_%pid%_1631901774729_ffappsvrprocess.log`, the placeholder is found 16 characters in (after `client_ffappsvr_`). The `sizeof(WCHAR)` is probably twice as large as the characters in the string, which leads to the `len == 8` result.

The minus operator probably calculates number of characters instead of size. Not sure. Let's replace it with something more expressive - [wcslen](#) that is documented to return the number of characters:

```
len = wcslen(string) - wcslen(p);
```

I've tested it and it works fine.

Please review branch 3821c, revision 14429. I messed up the branch creation / co and pushed from the local branch directly to 3821c. I'll be more careful next time. Btw can you protect 3821c from me until I understand b2r?

#9 - 12/09/2022 05:30 AM - Greg Shah

Code Review Task Branch 3821c Revision 14429

The change is good, however you are missing the history entry.

#10 - 12/09/2022 05:32 AM - Galya B

Greg Shah wrote:

Code Review Task Branch 3821c Revision 14429

The change is good, however you are missing the history entry.

Just pushed it in revision 14430. B2r commit of a few lines of changes to a local branch is very slow. Is it normal?

#11 - 12/09/2022 05:37 AM - Greg Shah

I messed up the branch creation / co and pushed from the local branch directly to 3821c. I'll be more careful next time. Btw can you protect 3821c from me until I understand b2r?

1. Post a .diff of your proposed changes in Redmine OR use a task branch that is created specifically off of 3821c.
2. Unbind (b2r unbind) your local copy of 3821c until it is time to commit changes. Any accidental commit will then just be local and won't update the central repo.

B2r commit of a few lines of changes to a local branch is very slow. Is it normal?

In your current setup, you are just using a local copy **bound to** the central repo. Any local commit in a "checkout" is modifying the central copy. As noted above, you can unbind to disconnect.

#12 - 12/09/2022 05:38 AM - Greg Shah

- Status changed from Review to Test
- % Done changed from 0 to 100

When you believe that all functional requirements of the task are complete, please set the % Done to 100%.

#13 - 12/09/2022 05:41 AM - Galya B

Greg Shah wrote:

In your current setup, you are just using a local copy **bound to** the central repo. Any local commit in a "checkout" is modifying the central copy. As noted above, you can unbind to disconnect.

I got it, it was just too late. The guide has way too many words and I get lost in it. The create command needs to be executed on the remote server instead on localhost and I didn't get that initially. In git you can create it locally and push/create to the remote.

#14 - 12/12/2022 01:28 PM - Galya B

- File Screenshot from 2022-12-12 20-26-50.png added

I ran the server on Windows with a batch process and was able to confirm the new behavior of the spawner of properly replacing pid.

#15 - 12/12/2022 04:34 PM - Greg Shah

- Status changed from Test to Closed

#16 - 02/27/2023 08:40 AM - Greg Shah

- Related to Feature #7147: Make FWD log outputs consistent added

Files

Screenshot from 2022-12-12 20-26-50.png

50.8 KB

12/12/2022

Galya B