

---

Subject: Added SysExec package

Posted by [mdelfede](#) on Sun, 25 May 2008 07:49:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

A small package with some SysExec() functions to launch external commands and gather their output and error data.

Not the state of the art, but handy sometimes

The provided functions are :

```
bool SysExec(String const &command, String const &args, const VectorMap<String, String>
&Environ, String &OutStr, String &ErrStr);
bool SysExec(String const &command, String const &args, String &OutStr, String &ErrStr);
bool SysExec(String const &command, String const &args, const VectorMap<String, String>
&Environ, String &OutStr);
bool SysExec(String const &command, String const &args, String &OutStr);
bool SysExec(String const &command, String const &args);
```

Parameters are :

command : command executable name  
args : a line of arguments, space separated  
Environ : a VectorMap containing environment data to pass  
OutStr : a reference to a string that will contain command output  
ErrStr : a reference to a string that will contain command error output

Commands will return true on success, false otherwise.

Caveats :

- 1- it will block application if launched command hangs
- 2- no timeout provided... but could be easily added
- 3- no mean to have some 'progress' indication of running app, but could be easily added

Path will be searched for command, so a complete path is not necessary.  
It "should" work on windows too, but I didn't test it yet.

Ciao

Max

---

---

Subject: Re: Added SysExec package  
Posted by [koldo](#) on Sat, 09 Aug 2008 20:29:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello mdelfede

Thank you for your functions. I am really interested in them plus:

- A version that could also write characters to the command
- A function to open a file looking first for the right program

For the first options I have found in the forums these two posts:

- [http://www.ultimatepp.org/forum/index.php?t=msg&goto=16018&&srch=CreateHostRunDir#msg\\_16018](http://www.ultimatepp.org/forum/index.php?t=msg&goto=16018&&srch=CreateHostRunDir#msg_16018)

```
StringStream ss;
```

```
Id id;
```

```
id.CreateHostRunDir()->Execute("valgrind --help", ss);
```

```
String txt = ss;
```

- [http://www.ultimatepp.org/forum/index.php?t=msg&goto=15620&&srch=StartProcess#msg\\_15620](http://www.ultimatepp.org/forum/index.php?t=msg&goto=15620&&srch=StartProcess#msg_15620)

```
String ret; // <-- Added this line
String cmdline = "ping -n 3 192.168.0.1";
One<SlaveProcess> sp = startprocess(cmdline);
while(sp->IsRunning()){
    Sleep(200);
    sp->Read(ret); // <-- Added this line
}
return sp->GetExitCode();
```

(The first post is from you)

It seems inside "The Ide" are the ideas to complete SysExec(). Please tell me were you are in their developing and if I can help to have them read as soon as possible.

Best regards

---

---

Subject: Re: Added SysExec package  
Posted by [Mindtraveller](#) on Fri, 31 Oct 2008 08:30:13 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Is it OK that SysExec("xterm") is successful, but SysExec("gzip") fails?

OS: FreeBSD 6.2

---

---

Subject: Re: Added SysExec package

Posted by [mdelfede](#) on Fri, 31 Oct 2008 18:04:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Mindtraveller wrote on Fri, 31 October 2008 09:30: Is it OK that SysExec("xterm") is successful, but SysExec("gzip") fails?

OS: FreeBSD 6.2

Uhmhhh... it shouldn't fail, but it depends on whether gzip is inside path. Can you try the Error variant and look at the resulting string ?

Max

For Kodo : sorry, I didn't notice your post on August

I made SysExec just because I needed one without having to link ide (CreateHostRunDir and so needs ide code....) and because I think that some system calls are handy even for non gui apps. I haven't developed them anymore, but if you want to add some features you're wellcome ! Just on these days I needed some way to call an external app redirecting input and output streams, and having some way for adding a Progress bar to it....

Max

---

---

Subject: Re: Added SysExec package

Posted by [koldo](#) on Fri, 31 Oct 2008 22:38:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Sorry mdelfede

From then I have done some functions. Some of them are in <http://www.ultimatepp.org/forum/index.php?t=msg&th=3942&> .

Between them I have included these functions:

```
// LaunchFile("Sheet.xls") will open the file with the program assigned by default to open xls files
bool LaunchFile(const String& file);
```

```
// LaunchCommand("mplayer myclip.avi", MyCallback) will launch the command line program with
args without opening a window, and all the output will be sent to MyCallback(String &) function
int LaunchCommand(const char *cmd, void (*readCallBack)(String &));
```

```
// LaunchCommand("mplayer myclip.avi",str) will do the same but sending the output to String str
```

```
int LaunchCommand(const char *cmd, String &ret);
```

The LaunchCommand functions are based in the second sample code included in the August post.

I am using them extensively in funtions working with mplayer, mencoder, ffmpeg, sox and other for a program that handles video and audio (see a screenshot in the post I have included above).

These functions parse the output of the command by "readCallBack" that handles a progress bar and a command window (in the screenshot), this one just for debugging, but goes very well. The functions work well in Windows and Linux.

I want to do a higher level class to handle extern processes in a very simple way and with progress handling included.

Best regards  
Koldo

---

Subject: Re: Added SysExec package  
Posted by [mdelfede](#) on Sat, 01 Nov 2008 14:39:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Very handy functions, Kodo !  
Do you think we can merge them with SysExec package, or you're thinking to a new package ?

Max

---

Subject: Re: Added SysExec package  
Posted by [koldo](#) on Sun, 02 Nov 2008 11:37:04 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Hello Max

In fact the "LaunchCommand" functions have nothing new as they are fully based in <usvn/SlaveProcess.h>, class LocalProcess.

As they handle command line functions I would like them to:

- Return a String with all the output
- Use a callback function that will get Strings from the command output (possibly line by line as command functions usually output this way). Gui version will have ProcessEvents() inside so that the program would be responsive.
- Now both wait for the command to end, so there could be other with a callback that gets a String

and a handle to check if the command has ended, to parse the String and send input to the command or to kill the command. These callback functions would be called from a Timer.

- Have the possibility of a command window for debugging

LaunchFile and GetExtExecutable (this one gets the default program that opens the provided extension) are different, but now they have not clear location (see post [http://www.ultimatepp.org/forum/index.php?t=msg&goto=18916&#msg\\_18916](http://www.ultimatepp.org/forum/index.php?t=msg&goto=18916&#msg_18916)).

I do not have any preferences as I am here only from August (but I will be here many time), so you have much more experience about how to organize Upp and to declare the functions. I will arrange them were you prefer.

Best regards  
Koldo (with "I")

---

---

Subject: Re: Added SysExec package  
Posted by [mdelfede](#) on Sun, 02 Nov 2008 19:58:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Well.... maybe it would be handy to have a Process class that encapsulates all that.... Maybe joined with my old good SysExec.

There's also something in my new Signal class that would belong to process handling....

I see clearly 2 new classes to arrange all that stuffs.

One for process handling, like stuffs that are now in SysExec and your execute stuffs, one for Signals, we should look for some way of signaling for windows too.... The third is already there, I see SystemLog stuffs to belong to the same group.

So :

Signal()  
SystemLog()  
Process()

In Process I'd put your code and the part of my Signal() stuff that checks/handles other app's instances.

I'd prefer these classes to not depend on Usvn, either....

Ciao

Max

---

---

Subject: Re: Added SysExec package

Posted by [koldo](#) on Mon, 03 Nov 2008 16:40:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello Max

Very interesting.

Question: What is Signal() and SystemLog()?

Could also send a draft of the declaration of these classes with a small explanation per method and variable, at least their "public:" side?

Best regards

Koldo

---

---

Subject: Re: Added SysExec package

Posted by [mdelfede](#) on Wed, 05 Nov 2008 21:45:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hy Koido!

Both Signal and SystemLog classes are documented in Bazaar.

Signal is just for Linux, maybe we could find some kind of inter process signaling for windows also...

Max

---

---

Subject: Re: Added SysExec package

Posted by [captainc](#) on Wed, 05 Nov 2008 23:38:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I've used this for signals on Windows before:

```
#include <signal.h>
bool run=true;
void sighandler(int sig)
{
    run=false;
};
CONSOLE_APP_MAIN
{
    signal(SIGABRT, sighandler); // register signal, if it is hit, call sighandler function
    signal(SIGINT, sighandler);
    signal(SIGTERM, sighandler);

    while(run)
    {
```

```
    // my code logic
}
}
```

Maybe provide a wrapper around singal.h or look at it's source, use it as a model, and modify to suit Upp.

---

---

Subject: Re: Added SysExec package  
Posted by [mdelfede](#) on Thu, 06 Nov 2008 14:49:17 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

captainc wrote on Thu, 06 November 2008 00:38I've used this for signals on Windows before:

```
#include <signal.h>
bool run=true;
void sighandler(int sig)
{
    run=false;
};
CONSOLE_APP_MAIN
{
    signal(SIGABRT, sighandler); // register signal, if it is hit, call sighandler function
    signal(SIGINT, sighandler);
    signal(SIGTERM, sighandler);

    while(run)
    {
        // my code logic
    }
}
```

Maybe provide a wrapper around singal.h or look at it's source, use it as a model, and modify to suit Upp.

Well, very interesting... I didn't know Windows provided quite a similar signal mechanism as Posix. I could try to make it fit to my Signals package.....

Max

---

---

Subject: Re: Added SysExec package  
Posted by [Mindtraveller](#) on Thu, 06 Nov 2008 15:22:32 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

mdelfede wrote on Fri, 31 October 2008 21:04Mindtraveller wrote on Fri, 31 October 2008 09:30Is it OK that SysExec("xterm") is successful, but SysExec("gzip") fails?

OS: FreeBSD 6.2

Uhmhhh... it shouldn't fail, but it depends on whether gzip is inside path. Can you try the Error variant and look at the resulting string ?

I tried and both out and error strings were empty. It worked only when I specified exact path to gzip. It is strange because in xterm, gzip is called (found I mean) without specifying exact path.

---

---

Subject: Re: Added SysExec package

Posted by [koldo](#) on Thu, 06 Nov 2008 17:00:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello Max

Sorry!. I have readed the SystemLog and SystemLogTest and I do not understand Signal() and SystemLog().

Also I would need rather urgently a SysExec version that instead of waiting for the extern program to end and returning a String with the output, it would let not to stop the main program and retrieve the output periodically to the main program in any way.

Best regards

Koldo (with "I" (not "i"))

---

---

Subject: Re: Added SysExec package

Posted by [mdelfede](#) on Tue, 11 Nov 2008 12:40:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

koldo wrote on Thu, 06 November 2008 18:00Hello Max

Sorry!. I have readed the SystemLog and SystemLogTest and I do not understand Signal() and SystemLog().

Also I would need rather urgently a SysExec version that instead of waiting for the extern program to end and returning a String with the output, it would let not to stop the main program and retrieve the output periodically to the main program in any way.

Best regards

Koldo (with "I" (not "i"))

Hi koLdo

I've found an idea with pipes to have a sysexec as you like... maybe next days I'll code it.



It would also be useful to have some progress indication of app running.

About SystemLog, there are some docs inside... did you see them ?

Max

---

Subject: Re: Added SysExec package  
Posted by [koldo](#) on Thu, 13 Nov 2008 11:23:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello Max

It is Ok. SystemLog is useful for me and I will use it in the Sys functions I am doing now.  
One question: How to send the log to certain file?

I have tried SysExec but compiling with MSC9 I get a Window error when executing  
`result = _spawnvpe(_P_WAIT, command, argv, envv);`

with this:

"Microsoft Visual C++ Debug Library. Program: ... File: ..\spawnvpe.c Line: 84 Expression: \*\*argv  
!= \_T('\0') ...

The program is:

```
String out;  
SysExec("mencoder", "", out);  
Compiling with MinGW the program works well.
```

Previously I had to change  
`bool SysExec(String const &command, String const &args, String OutStr)`  
{  
 String ErrStr;  
 return SysExec(command, args, Environment(), OutStr, ErrStr);  
}  
} as the third argument was declared in the SysExec.h file as "String &OutStr".

Ah!, I have used SVN 625.

Best regards  
Koldo

---

Subject: Re: Added SysExec package  
Posted by [mdelfede](#) on Fri, 14 Nov 2008 15:51:48 GMT

---

koldo wrote on Thu, 13 November 2008 12:23Hello Max

It is Ok. SystemLog is useful for me and I will use it in the Sys functions I am doing now.  
One question: How to send the log to certain file?

I have tried SysExec but compiling with MSC9 I get a Window error when executing  
result = \_spawnvpe(\_P\_WAIT, command, argv, envv);

with this:

"Microsoft Visual C++ Debug Library. Program: ... File: ..\spawnvpe.c Line: 84 Expression: \*\*argv  
!= \_T('\0') ...

The program is:

```
String out;  
SysExec("mencoder", "", out);  
Compiling with MinGW the program works well.
```

Previously I had to change  
bool SysExec(String const &command, String const &args, String OutStr)  
{  
 String ErrStr;  
 return SysExec(command, args, Environment(), OutStr, ErrStr);  
}  
} as the third argument was declared in the SysExec.h file as "String &OutStr".

Ah!, I have used SVN 625.

Best regards  
Koldo

It must be &OutStr, as is an output parameter, it gather execution command output. If you set  
without the &, it's passed as a value parameter.....

Max

---

Subject: Re: Added SysExec package  
Posted by [koldo](#) on Wed, 19 Nov 2008 07:05:35 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Sorry Max

I did the change in the declaration as in SysExec.cpp it is defined as:

```
Quote:bool SysExec(String const &command, String const &args, String OutStr)
{
    String ErrStr;
    return SysExec(command, args, Environment(), OutStr, ErrStr);
} // END SysExec()
Best regards
Koldo
```

---

---

Subject: Re: Added SysExec package  
Posted by [mdelfede](#) on Wed, 19 Nov 2008 08:44:54 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

koldo wrote on Wed, 19 November 2008 08:05Sorry Max

I did the change in the declaration as in SysExec.cpp it is defined as:

```
Quote:bool SysExec(String const &command, String const &args, String OutStr)
{
    String ErrStr;
    return SysExec(command, args, Environment(), OutStr, ErrStr);
} // END SysExec()
Best regards
Koldo
```

It was my mistake... The correct one is

```
bool SysExec(String const &command, String const &args, String &OutStr)
```

Ciao

Max

---

---

Subject: Re: Added SysExec package  
Posted by [mdelfede](#) on Thu, 20 Jan 2011 23:13:45 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

New functions to SysExec package (they were needed for Updater one) :

SysStart() series, starts an app without waiting for its completion

SysExecAdmin() series starts an app as the root user (linux) or admin user (windows) waiting for its completion. On linux you must provide a password, on windows it shows the security dialog

before proceeding.

By now, the windows version DON'T wait for completion, I must still finish it

SysStartAdmin() same as above, don't wait for completion.

SysExecUser()

SysStartUser()

Same as above, for a given user (linux). On windows, they fall back to SysExec() and SysStart().

The most important are the xxxAdmin() functions, which allow to start administrative tasks on Vista/Windows7 too.

Latest 2 allows app to go back to normal user on Linux; on windows I don't know if it's possible.

Ciao

Max

---

---

Subject: Re: Added SysExec package

Posted by [mdelfede](#) on Sun, 23 Jan 2011 17:36:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Added SysExecGui package, whith gui frontend for password reading (in Linux) and same functions (without password reading) for windows, to have an uniform interface.

It uses SysExec package, adding CtrlLib dependency for gui apps; for non-gui apps, use SysExec which don't have CtrlLib dependencies.

Still no docs, but code well commented.

Ciao

Max

---

---

Subject: Re: Added SysExec package

Posted by [dolik.rce](#) on Mon, 24 Jan 2011 08:54:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

mdelfede wrote on Sun, 23 January 2011 18:36Added SysExecGui package, whith gui frontend for password reading (in Linux) and same functions (without password reading) for windows, to have an uniform interface.

It uses SysExec package, adding CtrlLib dependency for gui apps; for non-gui apps, use SysExec which don't have CtrlLib dependencies.

Hi Max,

This is a very nice package and definitely useful. I have just one proposal: what about merging SysExec and SysExecGui into one package? I think it makes things easier for user. All you would need to do is:

- 1) Move files from SysExecGui to SysExec
- 2) Put `#ifdef flagGUI` guard around the code in SysExecGui.cpp
- 3) Put `#include SysExecGui.h` at the end of SysExec.h (again only `#ifdef flagGUI`)
- 4) Add package CtrlLib into SysExec with "when GUI" condition
- 5) Check the include paths for SysExecGui leftover and few other simple to solve details.

It would make life a little bit easier for people (like me) who sometimes write apps that can be compiled both with GUI and CLI or who start with CLI and later switch to GUI. But it is just a proposal, if you think it is not a clean solution feel free to leave it as is, after all it is not that hard to make it work the same way on the user package level

Best regards,  
Honza

---

---

Subject: Re: Added SysExec package  
Posted by [mdelfede](#) on Mon, 24 Jan 2011 16:51:20 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Honza,

I can think about, it seems a nice solution

I'm still working on the package, Win\$ Vista and later are a nightmare for system programming. The (useless) protection scheme that they introduced is always on the way. I hope not to have problems on Protect package too.... I'm very afraid about it.

Ciao

Max

---

---

Subject: Re: Added SysExec package  
Posted by [luoganda](#) on Sat, 29 Jul 2017 13:57:02 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

```
SysStart("notepad", "\"D:\\zz_download\\test.bat\"");
```

This gives unhandled exception, any ideas?

---