
Subject: Re: Upp 2007.1beta3 released
Posted by [piotr5](#) on Tue, 13 Feb 2007 11:45:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

except for the already mentioned bugs, it just works great.
for the interrupted compilation I have found the solution,
I think:

```
--- upp-uvs/Web/sproc.cpp 2007-01-06 14:59:48.000000000 +0100
+++ test/Web/sproc.cpp 2007-02-13 03:11:48.000000000 +0100
@@ -111,7 +111,7 @@
    if(rpipe[1] >= 0) { close(rpipe[1]); rpipe[1] = -1; }
    if(wpipe[0] >= 0) { close(wpipe[0]); wpipe[0] = -1; }
    if(wpipe[1] >= 0) { close(wpipe[1]); wpipe[1] = -1; }
- waitpid(pid, 0, WNOHANG | WUNTRACED);
+ if(pid) waitpid(pid, 0, WNOHANG | WUNTRACED);
    pid = 0;
    output_read = false;
#endif
@@ -328,7 +328,7 @@
    kill(pid, SIGTERM);
    GetExitCode();
    int status;
- if(waitpid(pid, &status, 0) == pid)
+ if(pid && waitpid(pid, &status, 0) == pid)
    DecodeExitCode(status);
    exit_string = "Child process has been killed.\n";
}
```

and maybe

```
--- upp-uvs/ide/Host.cpp 2007-01-06 14:56:44.000000000 +0100
+++ test/ide/Host.cpp 2007-02-13 03:11:29.000000000 +0100
@@ -158,7 +158,7 @@
    Vector<int>& pid = sPid();
    int i = 0;
    while(i < pid.GetCount())
- if(waitpid(pid[i], 0, WNOHANG | WUNTRACED) > 0)
+ if(pid[i] && waitpid(pid[i], 0, WNOHANG | WUNTRACED) > 0)
    pid.Remove(i);
    else
    i++;
```

the point of this bugfix is that if one keeps calling waitpid
with an explicite pid, then using the pid "0" should be
avoided at all cost, as it does act in an unexpected way:

instead of waiting for the process with pid 0, it just waits for any process with the same group-id as the calling program, and thereby the exitcode could get lost...
