
Subject: SetEnv()
Posted by [koldo](#) on Tue, 11 Sep 2012 08:28:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek

I could not find a setenv function. This could be an implementation:

```
bool SetEnv(const char *id, const char *val)
{
#ifdef PLATFORM_POSIX
    return setenv(id, val, 1) == 0;
#else
    return _wputenv(WString(id) + "=" + WString(val)) == 0;
#endif
}
```

Subject: Re: SetEnv()
Posted by [mirek](#) on Tue, 11 Sep 2012 11:05:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

I am afraid it is not so simple, consider

```
const VectorMap<String, String>& Environment();
```

I am afraid that adding it in this form could cause some confusion...

Subject: Re: SetEnv()
Posted by [koldo](#) on Tue, 11 Sep 2012 11:43:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek

Environment() returns environment vars and values.

You can just choose a right name and interface for a function that sets an environment var.

Subject: Re: SetEnv()
Posted by [mirek](#) on Tue, 11 Sep 2012 15:13:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sorry for not being clear enough. Problem I see is that Environment returns a reference to environment variables.

If you use SetEnv, you might expect this to change, which in practice is quite difficult to achieve (e.g. because of MT issues).

Mirek

Subject: Re: SetEnv()
Posted by [koldo](#) on Tue, 11 Sep 2012 22:26:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek

In my case the program launches another program modifying previously PATH and adding some new variables.

I thought that this solution was easier than changing Sys() and LocalProcess functions to set environment variables for child programs.

Initially if somebody uses Environment(), has to be aware that using SetEnv() will change variables so this does not seem to be a problem.

About MT, I am not sure if environment variables are MT safe at least in Windows. Perhaps SetEnv() would have to be MT protected.

Subject: Re: SetEnv()
Posted by [mirek](#) on Wed, 12 Sep 2012 06:56:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

koldo wrote on Tue, 11 September 2012 18:26

About MT, I am not sure if environment variables are MT safe at least in Windows. Perhaps SetEnv() would have to be MT protected.

Once again, that is not what I was addressing. The problem is that I would want to fix Environment to reflect changes by SetEnv, then everything would be fine.

Unfortunately with current intefrace it is not quite possible...

Mirek

Subject: Re: SetEnv()
Posted by [koldo](#) on Wed, 12 Sep 2012 09:19:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek

Now I understand. U++ programs read environment vars at the beginning in `ApplnitEnvironment__()` and put a copy of them in `EnvMap()`.

This way a `SetEnv()` function should update `EnvMap()`. For example:

```
bool SetEnv(const char *id, const char *val)
{
    EnvMap().Put(WString(id), WString(val));
#ifdef PLATFORM_POSIX
    return setenv(id, val, 1) == 0;
#else
    return _wputenv(WString(id) + "=" + WString(val)) == 0;
#endif
}
```

However `EnvMap()` is not public.

Subject: Re: SetEnv()
Posted by [mirek](#) on Wed, 12 Sep 2012 09:45:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, but that is not so simple in MT...

Consider that one thread takes a reference and other then uses `SetEnv`.

Do not get me wrong, I think `SetEnv` is a good idea, but there are issues to consider and think through...
