Module
Process::GID

Provides a higher-level (and more portable) interface to the underlying operating system's concepts of real, effective, and saved group IDs. Discussing of the semantics of these IDs is well beyond the scope of this book: readers who want to know more should consult POSIX documentation or read the intro(2) man pages on a recent Unix platform. All these methods throw NotImplementedError if the host operating does not support a sufficient set of calls. The descriptions that follow are based on notes in ruby-talk:76218 by Hidetoshi Nagai.

Module methods

change_privilege   Process::GID.change_privilege(*gid*) → *gid*

Sets the real, effective, and saved group IDs to *gid*, raising an exception on failure (in which case the state of the IDs is not known).

This method is not compatible with Process.gid=.

eid   Process::GID.eid → *egid*

Returns the effective group ID for this process. Synonym for Process.egid.

eid=   Process::GID.eid = *egid*

Synonym for Process::GID.grant_privilege.

grant_privilege   Process::GID.grant_privilege(*egid*) → *egid*

Sets the effective group ID to *egid*, raising an exception on failure. One some environments this may also change the saved group ID (see re_exchangeable?).

re_exchange   Process::GID.re_exchange → *egid*

Exchanges the real and effective group IDs, setting the saved group ID to the new effective group ID. Returns the new effective group ID.

re_exchangeable?   Process::GID.re_exchangeable → true or false

Returns true if real and effective group IDs can be exchanged on the host operating system and returns false otherwise.

rid   Process::GID.rid → *gid*

Returns the real group ID for this process. Synonym for Process.gid.

sid_available?   Process::GID.sid_available? → true or false

Returns true if the underlying platform supports saved group IDs and returns false otherwise. Currently, Ruby assumes support if the operating system has setresgid(2) or setegid(2) calls or if the configuration includes the POSIX_SAVED_IDS flag.

switch   Process::GID.switch → *egid*
                                    Process::GID.switch { *block* } → *obj*

Handles the toggling of group privilege. In the block form, automatically toggles the IDs back when the block terminates (but only if the block doesn't use other calls into Process::GID calls, which would interfere). Without a block, returns the original effective group ID.