MiscRemoteSubprocess

Inherits From: MiscSubprocess

Declared In: <misckit/MiscRemoteSubprocess.h>

Class Description

This subclass of MiscSubprocess allows a process to be executed on another host. A command may be set up and executed as it would be for any regular MiscSubprocess. To change to another host besides the local host, simply use the ±**setHost:** method. Use ±**host** to find out upon which host a command will execute.

As a convenience, the ±**execute:onHost:** method will set up a host for execution and then execute the command. Also for the sake of convenience, the name of the local host may be determined by the +**thisHost** class method.

Because of slight differences between running processes remotely versus locally, the MiscSubProcess ±**pause:** and ±**resume:** methods do not affect a remote process. The ±**terminate:** method does, however, function. It should be noted that the UNIX$^{\text{TM}}$ `rsh` command is used to execute processes remotely.

Instance Variables

id **host**;
int **remoteExitCode**;

host — MiscString with the name of the host where commands are to be executed.

remoteExitCode — Currently unused.

Method Types

Executing a Command	± execute:onHost:
Finding theLocal Host	+ thisHost
Setting the Host for Execution	± host ± setHost:

Class Methods

thisHost

+ (const char *)**thisHost**

Returns a string pointing to the name of the host on which the application is being run.

See also: ±**host**

Instance Methods

execute:onHost:

- **execute:**(const char *)*aString*
  onHost:(const char *)*hostname*

Executes the command *aString* on the host *hostname*.

See also: ±**setHost:**

host

- (const char *)**host**

Returns the name of the host upon which the subprocess will execute, if any.

See also: +**thisHost** and ±**setHost:**

setHost:

- **setHost:**(const char *)*aString*

Sets the name of the host upon which the subprocess will execute to *aString*.

See also: ±**execute:onHost:** and ±**host**