

Sending Commands to SIMetrix/SIMPLIS

Overview

This document describes how to use the SxCommand utility to send command to SIMetrix from a remote program. Note that much of the functionality of SxCommand is now available from the sxdif.dll data interface Dll. See DataInterface.pdf for details. This can be found at <https://www.simetrix.co.uk/supplementary/Script/Remote-Access/RemoteDataInterface/docs/DataInterface.pdf>.

Sending Commands

SIMetrix employs an inter-process communication (IPC) technique that it uses to pass data between the simulator and front-end as well as providing a means for external programs to send commands.

To avoid the external program having to implement the IPC method directly, a special utility has been developed that sends commands to a running instance of SIMetrix. The utility can also optionally start a SIMetrix process if one is not already running.

Installation

The SxCommand utility is provided as part of the standard SIMetrix install and is located in the same directory as the main SIMetrix executable binaries. This location should not be changed as the – launch feature will not function if it is located anywhere else.

SxCommand Syntax

```
SxCommand [-sync timeout] [-immediate] [-ident ident] [-launch] [-delay delay]  
[-quiet] command
```

Where:

- sync *timeout* If specified, SxCommand will not return until SIMetrix completes the command sent or after the specified timeout period whichever occurs first. *timeout* is in units of **0.1 seconds**. A value of zero indicates no timeout - i.e. SxCommand will not return until SIMetrix completes its command or terminates.
- immediate If specified, the command is acted on immediately even if another operation is in progress. This will only work with a few commands and is intended to be used to send commands to cancel operations in force.
- launch If specified, will start a SIMetrix process if one is not already running.
- delay *delay* Only effective if used with -launch. Introduces a delay between launching SIMetrix and attempting to connect to it. *delay* is in milliseconds.
- ident *ident* This provides a means of communicating with more than one instance of SIMetrix if required. *ident* should match the ident value used to launch SIMetrix either using

“SxCommand -launch -ident *ident*” or by starting SIMetrix directly with “SIMetrix /ident *ident*”.

-quiet No error messages will be returned if this is specified. Otherwise any errors will be reported on stderr.

command SIMetrix command to be executed. This may be a primitive command or a script. If the command contains spaces the whole command string should be enclosed in double quotation marks. If double quotation marks are required in the command itself, they should be preceded by a back slash character. Backslash characters not followed by a double quote are interpreted literally - there is no need to double them up.

SxCommand will return the following values according to its success or otherwise:

- 0 Command sent successfully.
- 1 Error: could not connect to SIMetrix process
- 2 Error: incorrect arguments
- 3 Error: SIMetrix failed to launch

The following error codes will only be returned if the -sync switch is specified:

- 4 Error: timeout elapsed
- 5 Error: Command failed
- 6 Error: SIMetrix is busy
- 7 Error: SIMetrix connection failed before command completed

If -quiet is not specified, an appropriate error message will also be sent to stderr if an error occurs.

Calling SxCommand From a ‘C’ Program

The following code samples show one method of calling SxCommand from another program.

Example Code

```
#include <stdio.h>
#include <sys/types.h>
#include <errno.h>
#include <process.h>
char *command = "\"Plot vout\"" ;
char *sxCommand = "C:\\Program Files\\SIMetrix50\\SxCommand.exe" ;
int main()
{
    char *args[] = {sxCommand, command, NULL} ;
    execv(sxCommand, args) ;
    perror("Failed to exec ") ;
    return 1 ;
}
```

Note that it is necessary to enclose the command in quotation marks. This is because the underlying API - CreateProcess - that launches a new process is only able to accept a single argument string. This string is parsed into multiple arguments by the process itself and quotation marks are used to

delimit arguments that contain white space. To send a literal quotation mark you must prefix it with a backslash.

Testing

For testing purposes, the SIMetrix “Echo” command may be useful. It simply displays the supplied text in the command shell’s message window. You could use:

```
char *command = "Echo \\\"Hello World!\\\" " ;
```