



Adding Processes to the Subsystem Resource Controller for Better Control of your own Daemons

*Contributed by Mark Neuman
Manager of Technical Solutions
Datatrend Technologies, Inc.*

The Subsystem Resource Controller (SRC) is a master daemon control mechanism within the AIX Operating System. It allows you, and the system, to control other registered daemons and provides an easy interface for managing those processes. The AIX administrator should be familiar with the "startsrc" and "stopsrc" commands which are two control point commands for managing processes registered with the SRC subsystem.

You, as the system administrator, may register programs (or shell scripts) with the SRC subsystem. This will give you much better control over the behavior of these processes. For example, you can elect to have only one copy of the daemon running at a time and the SRC subsystem will do the checking for you. In addition, you can tell the SRC subsystem to respawn the process if it abnormally terminates. This will protect you in case the daemon dies or is killed outside of the SRC subsystem. Signal handling is processes via the SRC subsystem; therefore you will have to code traps to respond to certain signals in order to gracefully stop when requested. Once registered, the process can be started with a "startsrc -s name" command so you do not need to remember the path to your daemon.

As you create a daemon program, or script, you need to pay attention to the signals that will be used for control within the subsystem. The primary signals are the TERM (signal 15) or the KILL (signal 9). Functions can be added to provide for a clean shutdown when one of these signals is sent, as exemplified below.

For a Clean Shutdown	
	#!/bin/ksh
	trap "" HUP INT QUIT TSTP STOP
	trap "shutdown_ss" TERM
	shutdown_ss ()
	{
	#cleanup stuff
	exit ()
	}
	while true
	do
	#your stuff
	done

The command used to "register" your new subsystem is the mkssys command. However, there are a number of options associated with this command.

Common mkssys command options

Flag(s)	Description
-s	Name of the new subsystem
-p	Full path to executable program / shell script.
-u	Numeric uid for subsystem execution. Must be a defined user.
	<i>Optional information:</i>
-i, -o, -e	Standard input, output, error devices. Default is /dev/console
-a	Arguments to the executable.
-G	Group name for related subsystems.
-K, -S, -I	Method of communications (Sockets [-K], Signals [-S], or Messages [-I]). Default is Sockets.
-n, -f	The signal numbers used for normal (-n), or forced (-f), subsystem stop, if signals are used for communications.
-E	"Nice" value for subsystem execution priority.
-R, -O	Automatic restart (-R), or abort (-O) on abnormal termination. Whether to respawn or not.
-Q, -q	Single (-Q), or multiple (-q) instances allowed.
-w	Wait time between a normal and forced stop.

An example of registering a shell script into the Subsystem Resource Controller is given below:

mkssys
-s example \
-p /usr/local/sample/script.sh \
-u \$(uid -u user) \
-S \
-n 15 \
-f 9 \
-R \
-Q

In the above example:

1. The "-s" argument defines the subsystem "name" (startsrc -s name) would be example, substitute your own names there and they must be unique.
2. The "-p" is the path to the executable, and in this example the executable object is script.sh and the path is /usr/local/sample.
3. The "\$ (uid -u user)" part of the "-u \$(uid -u user)" option line will determine the uid for the user "user" and substitute that, otherwise the "-u" argument requires a uid number.
4. The "-S" argument tells the SRC to use signals to communicate.
5. The "-n 15" defines signal 15 (SIGTERM) to be the normal stop signal.
6. The "-f 9" defines the signal 9 (SIGKILL) to be the forced stop signal.
7. The "-R" tells the SRC to restart the process if it abnormally terminates. That is, if the process dies without the SRC system telling it to.
8. The "-Q" will have the SRC system enforce having only one instance active at a time.

After registering your new subsystem "example" you can:

1. Start it with the "startsrc -s example" command.
2. Stop it with the "stopsrc -s example" command.
3. Check on it with the "lssrc -s example" command.

You can use the "chssys" command to make changes to the defined subsystem and "rmssys" to remove the subsystem entirely. Please be very careful with these commands and do not make changes or remove system subsystems as you could lose your OS functionality. The definitions for this and all Subsystem Resource Controller subsystems are in the SRCsubsys object class within the ODM system, as with all other ODM databases look but do not touch unless you both know what you are doing AND have a current mkysyb backup.

Datatrend's TrendSetter eNewsletter
October 15, 2002