How can I trap interrupts in my shell script?

How can I trap interrupts in my shell script?

On this page:

Overview
tcsh
bash
Example scripts

Overview

Both bash and tcsh allow you to trap interrupts (generated by pressing Ctrl-C) in your shell scripts and deal with them rather than simply exiting. This can be useful in several cases: your script may generate some temporary files which you would like to clean up before exiting, or you may wish to disable interrupts during a series of tasks to ensure the user doesn't inadvertently quit your script and leave their files in a broken state.

A word of caution: when testing this functionality, ensure that you have another way to end your script should it become unresponsive.

tcsh

The C shell provides a mechanism for trapping interrupts generated by Control-C with the onintr statement.

There are three forms:

`onintr label`	Go to `label` on interrupt
`onintr -`	Ignore all interrupts (Use with caution)
`onintr`	Restore interrupts to default.

See the example below for more detail.

bash

The Bourne shell provides a mechanism for trapping many "signals" sent to your process. One of the signals is the "INT" (interrupt) signal, which is generated when you press Ctrl-C.

`trap 'echo "Shell command here";' INT`	Perform any shell command in the single quotes when the "INT" signal is generated
`trap '' INT`	Ignore "INT" signal (use with caution)
`trap - INT`	Restore the "INT" signal handler to the default action

See the example below for more detail.

Example scripts

An example tcsh script:

```
#!/bin/athena/tcsh

onintr int
echo "Counting to 5 slowly, press Ctrl-C to interrupt."
foreach number (1 2 3 4 5)
    echo $number
    sleep 1
end

onintr -
echo "Counting to 5 again, but pressing Ctrl-C shouldn't work."
foreach number (1 2 3 4 5)
    echo $number
    sleep 1
end

onintr
echo "One more time, but Ctrl-C should work again."
foreach number (1 2 3 4 5)
    echo $number
    sleep 1
end
exit 0

int:
    echo "Hey, you pressed Ctrl-C.  Time to quit."
    exit 1
```

An example bash script:

```
#!/bin/athena/bash

trap '{ echo "Hey, you pressed Ctrl-C.  Time to quit." ; exit 1; }' INT
echo "Counting to 5 slowly, press Ctrl-C to interrupt."
for number in 1 2 3 4 5; do
    echo $number
    sleep 1
done

trap '' INT
echo "Counting to 5 again, but pressing Ctrl-C shouldn't work."
for number in 1 2 3 4 5; do
    echo $number
    sleep 1
done

trap - INT
echo "One more time, but Ctrl-C should work again."
for number in 1 2 3 4 5; do
    echo $number
    sleep 1
done
exit 0
```