

Debugging

What You Will Learn

- Built-in Bash Options for Debugging

Why Debug?

- A bug is really an error.
- Examine the inner workings of your script.
- Determine the root of unexpected behavior.
- Fix bugs (errors).

Built in Debugging Help

- -x = Prints commands as they execute
- After substitutions and expansions
- Called an x-trace, tracing, or print debugging
- `#!/bin/bash -x`
- `set -x`
 - `set +x` to stop debugging

```
#!/bin/bash -x  
TEST_VAR="test"  
echo "$TEST_VAR"
```

```
+ TEST_VAR=test  
+ echo test  
test
```

```
#!/bin/bash
TEST_VAR="test"
set -x
echo $TEST_VAR
set +x
hostname
```

```
+ echo test
test
+ set +x
linuxsvr
```

Built in Debugging Help

- -e = Exit on error.
- Can be combined with other options.
 - `#!/bin/bash -ex`
 - `#!/bin/bash -xe`
 - `#!/bin/bash -e -x`
 - `#!/bin/bash -x -e`

```
#!/bin/bash -e  
FILE_NAME="/not/here"  
ls $FILE_NAME  
echo $FILE_NAME
```

```
ls: cannot access /not/here: No  
such file or directory
```



```
#!/bin/bash -ex  
FILE_NAME="/not/here"  
ls $FILE_NAME  
echo $FILE_NAME
```

```
+ FILE_NAME=/not/here  
+ ls /not/here  
ls: cannot access /not/here: No such  
file or directory
```

Built in Debugging Help

- -v = Prints shell input lines as they are read.
- Can be combined with other options.

```
#!/bin/bash -v  
TEST_VAR="test"  
echo "$TEST_VAR"
```

```
#!/bin/bash -v  
TEST_VAR="test"  
echo "$TEST_VAR"  
test
```

```
#!/bin/bash -vx  
TEST_VAR="test"  
+ TEST_VAR=test  
echo "$TEST_VAR"  
+ echo test  
test
```

For more information.

help set | less

Summary

- Built-in Bash Options
 - -X
 - -e
 - -v