A Quick Guide to Bash Shell Redirection

One of the powerful features of UNIX (and Linux in specific) is the ability to redirect input and output from one process to another.

The Bash shell contains special commands, known as redirectors, that make this process intuitive and easy.

Table D.1 Bash Redirectors

Symbol	Example	Description
¦	<c1> ¦ <c2>	Takes the standard output of <c1> and uses it as standard input for <c2>.
>	<c1> > <file>	Places the standard output of <c1> into the file <file>, erasing <file>'s contents in the process.
<	<c1> < <file>	Uses the file <file> as standard input for <c1>.
>>	<c1> >> <file>	Appends the standard output of <c1> to the end of the file <file>.
>¦	<c1> >¦ <file>	Places the standard output of <c1> into the file <file>, erasing <file>'s contents in the process, even if the noclobber flag is set.
<>	<c1> <> <file>	Uses <file> as both standard input and standard output for <c1>.
&> or >&	<c1> &> <file>	Redirects both standard output and standard error output to <file>. Note that >& is deprecated.
2>	<c1> 2> <file>	Redirects standard error output from command <c1> into <file>, overwriting <file> in the process.
2>>	<c1> 2>> <file>	Appends the standard error output from process <c1> at the end of <file>.
2>¦	<c1> 2>¦ <file>	Redirects standard error output from command <c1> into <file>, overwriting <file> in the process, even if the noclobber flag is set.