Wildcards

Wildcards may be used to specify a set of files on the command line:

```
% ls *.txt

jim.txt

bob.txt

%
```

In a shell environment wildcards are expanded on the command line, rather than within each command. This means that expanded filenames are treated as text:

```
% echo "These are my text files:" *.txt

These are my text files: jim.txt bob.txt

%
```

Wildcards may be matched within the current working directory or at the end of a pathname. Matching in the middle of a pathname is not supported:

Good wildcard patterns:

```
*.h

:tmp:*.h

"my disk:my folder:my *"
```

Bad wildcard patterns:

```
*:tmp.h

"my disk:*:my file"
```

Wildcard Summary

*

- Match zero or more characters. For example, b* matches the files "b", "bat", "belfry",

and so on.

? - Match exactly one character. For example b? matches "bb", "bz, and so on.

[abc] - Match any character listed in the brackets. For example b[cd] matches "bc" or "bd".

[a-d] - Match any character in the range "a" to "d". For example, b[w-z] matches "bw", "bx", "by" or "bz".

[^abc] - Match any character not listed in the brackets. For example b[!a-z] matches "b0" and "b2", but not "ba" or "bi".

* - Match a literal "*".

\? - Match a literal "?".

\[- Match a literal "[".

Sample Patterns

*.txt - Match all files which end with ".txt".

b* - Match all files which start with "b".

tt - Match all files which contain a "tt" anywhere.

[abc]* - Match all files which start with an "a" or "b" or "c".

[a-zA-Z0-9]* - Match all files which start with an alphanumeric character.

b*[0-9] - Match all files which start with a "b" and end with a number.

Wildcards and Quotes

Macintosh filenames often contain spaces. To allow matching on such names, the nShell allows quoting of wildcard specifications:

"my *" Match all files which start with "my" and a space.

Note that the entire string is treated as a pattern. The command:

```
% echo "These are my files: my *"
```

would attempt to match files which start with "These are my...". To list my files, I would

use the command:

```
% echo "These are my files:" "my *"
```

In this case, only the second parameter is expanded as a wildcard.

Single quotes suppress wildcard expansion, as shown below.

Suppressing Wildcard Expansion

Strings contained within single quotes are not expanded:

```
% echo 'my favorite pattern is *.txt'

my favorite pattern is *.txt
```

Backslash characters may be used to force single characters to be treated literally.