Information on file globbing in Ubuntu 12.04 (the OS on which Cloudbiolinux from June 2012 is based) is available by typing man glob(7) in a terminal window, or at the website

http://manpages.ubuntu.com/manpages/precise/man7/glob.7.html

Description

Long ago, in UNIX V6, there was a program /etc/glob that would expand wildcard patterns. Soon afterward this became a shell built-in. Note that wildcard patterns are not regular expressions, although they are a bit similar. First, they match filenames, rather than text; second, the conventions are not the same: for example, in a regular expression '*' means zero or more copies of the preceding thing.
Now that regular expressions have bracket expressions where the negation is indicated by a '^', POSIX has declared the effect of a wildcard pattern "[^...]" to be undefined.

Wildcard Matching

A string is a wildcard pattern if it contains one of the characters '?', '*' or '['. Globbing is the operation that expands a wildcard pattern into the list of pathnames matching the pattern. Matching is defined by:
A '?' (not between brackets) matches any single character.
A '*' (not between brackets) matches any string, including the empty string.

Character classes

An expression "[...]" where the first character after the leading '[' is not an '!' matches a single character, namely any of the characters enclosed by the brackets. **The string enclosed by the brackets cannot be empty; therefore ']' can be allowed between the brackets, provided that it is the first character.** Thus, "[][!]" matches the three characters ']', '[', and '!'. **If the first character after the leading '[' is an '!', the expression matches any single character that is not matched by the expression that follows the '!'.** For example, "[!]a-]" matches any single character except ']', 'a' and '-'. To match a '!' character, it must **not** be the first character after the opening '['.

There is one special convention: two characters separated by '-' denote a range. Thus, "[A-Fa-f0-9]" is equivalent to "[ABCDEFabcdef0123456789]". One may include '-' in its literal meaning by making it the first or last character between the brackets. Thus, "[]-]" matches just the two characters ']' and '-', and "[--0]" matches only the three characters '-', '.', '0', because '/' cannot be matched. What does '/' have to do with this? The range of characters is based on the American Standard Code for Information Interchange (ASCII) encoding, which lists characters in the following order: !"#$%&'()*+,-./ are followed by the digits 0 through 9, followed by the characters :;<=>?@ which are followed by the upper case letters from A to Z in the English alphabet. The upper case letters are followed by [\]^_` which are followed in turn by lower case letters from a to z in the English alphabet, followed by {|}~.

Character classes and Internationalization

Of course ranges were originally meant to be ASCII ranges, so that "[-%]" stands for "[!"#$%]" and "[a-z]" stands for "any lowercase letter". Some UNIX implementations generalized this so that a range X-Y stands for the set of characters with code between the codes for X and for Y. However, this requires the user to know the character coding in use on the local system, and moreover, is not convenient if the collating sequence for the local alphabet differs from the ordering of the character codes. Therefore,

POSIX extended the bracket notation greatly, both for wildcard patterns and for regular expressions. In the above we saw three types of items that can occur in a bracket expression: namely (i) the negation, (ii) explicit single characters, and (iii) ranges. POSIX specifies ranges in an internationally more useful way and adds three more types:
(iv) Ranges are modified so that X-Y comprise all characters that fall between X and Y (inclusive) in the current collating sequence as defined by the LC_COLLATE category in the current locale.

(v) Named character classes, like [:alnum:] [:alpha:] [:blank:] [:cntrl:] [:digit:] [:graph:] [:lower:] [:print:] [:punct:] [:space:] [:upper:] [:xdigit:], so that one can say "[[:lower:]]" instead of "[a-z]", and have things work in Denmark, too, where there are three letters past 'z' in the alphabet. These character classes are defined by the LC_CTYPE category in the current locale.

(vi) Collating symbols, like "[.ch.]" or "[.a-acute.]", where the string between "[." and ".]" is a collating element defined for the current locale. Note that this may be a multicharacter element.

(vii) Equivalence class expressions, like "[=a=]", where the string between "[=" and "=]" is any collating element from its equivalence class, as defined for the current locale. For example, "[[=a=]]" might be equivalent to "[a'a`a"a^a]" (warning: Latin-1 here), that is, to "[a[.a- acute.][.a-grave.][.a-umlaut.][.a-circumflex.]]".

The manual page refers to ‘POSIX’ – this acronym refers to a family of standards released by IEEE, and is short for “Portable Operating System Interface”. It defines standard interfaces for many aspects of Unix and Linux operating systems, but not all aspects of all Unix or Linux systems are necessarily POSIX-compliant.