**Maven Versioning**

**Dependency version specification:**

**1.0:** "Soft" requirement on 1.0, just a recommendation

**[1.0]:** "Hard" requirement on 1.0

**(,1.0]:** x <= 1.0

**[1.2,1.3]:** 1.2 <= x <= 1.3

**[1.0,2.0):** 1.0 <= x < 2.0

**[1.5,):** x >= 1.5

**(,1.0],[1.2,):** x <= 1.0 or x >= 1.2; multiple sets are comma-separated

**(,1.1),(1.1,):** this excludes 1.1 (Useful when you know this version of library doesn’t work with your project)

**Maven's Standard Versioning Scheme:**

MajorVersion . MinorVersion . IncrementalVersion . BuildNumber . Qualifier

Examples:

1.2-SNAPSHOT

1.4.2-12

1.2-beta-2

**Qualifiers?**

Non numeric tokens are called qualifiers.

Ex:

1.2-apple > 1.2-banana

**Few standard qualifiers:** "alpha" < "beta" < "milestone" < "rc" = "cr" < "snapshot" < "" = "final" = "ga" < "sp"

Ex:

1.2-beta > 1.2-alpha

".qualifier" < "-qualifier" < "-number" < ".number"

Ex:

1.2-alpha > 1.2.alpha

**Few points to note:**

* **All versions with a qualifier are older than the same version without a qualifier (release version).**

For example:

1.2 > 1.2-beta-2

* **Identical versions with different qualifier fields are compared by using basic string comparison.**

For example:

1.2-beta-2 is newer than 1.2-alpha-6.

* **Maven corrects the version, by splitting the version by “-“**

For example:

1.2-abc2gfd -> 1.2-abc-2-gfd

1.2 -> 1.2

abc2gfd -> abc-2-gfd

1.2-abc-.1 -> 1.2-abc-0.1

1.2 -> 1.2

abc -> abc

.1 -> 0.1

* **Trailing "null" values (0, "", "final", "ga") are trimmed**

For example:

1.0.0-abc.0.0 -> 1-abc

* 1. -> 1

abc.0.0 -> abc