**Setting up the cron jobs in Jenkins using “Build periodically” – scheduling the jenins Job**

**Examples –**  
To schedule your build every 5 minutes, this will do the job : \*/5 \* \* \* \* OR H/5 \* \* \* \*

To the job every 5min past every hour(5th Minute of every Hour) 5 \* \* \* \*

To schedule your build every day at 8h00, this will do the job : 0 8 \* \* \*

To schedule your build for 4, 6, 8, and 10 o’clock PM every day – 0 16,18,20,22 \* \* \*

To schedule your build at 6:00PM and 1 AM every day – 0 1,18 \* \* \*

To schedule your build start daily at morning – 03 09 \* \* 1-5

To schedule your build start daily at lunchtime – 00 12 \* \* 1-5

To schedule your build start daily in the afternoon – 00 14 \* \* 1-5

To schedule your build start daily in the late afternoon – 00 16 \* \* 1-5

To schedule your build start at midnight – 59 23 \* \* 1-5 OR @midnight

To run a job on 9.30p.m. (at night) on 3rd of May then I ll write or 21.30 on 3/5/2011 – 21 30 3 5 \*

Every fifteen minutes (perhaps at :07, :22, :37, :52) 0 – H/15 \* \* \* \*

Every ten minutes in the first half of every hour (three times, perhaps at :04, :14, :24) – H(0-29)/10 \* \* \* \*

Once every two hours every weekday (perhaps at 10:38 AM, 12:38 PM, 2:38 PM, 4:38 PM) – H 9-16/2 \* \* 1-5

Once a day on the 1st and 15th of every month except December – H H 1,15 1-11 \*

**CRON expression**

A CRON expression is a string comprising five or six fields separated by white space that represents a set of times, normally as a schedule to execute some routine.

**Format**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Mandatory?** | **Allowed values** | **Allowed special characters** | **Remarks** |
| Minutes | Yes | 0-59 | \* / , – | – |
| Hours | Yes | 0-23 | \* / , – | – |
| Day of month | Yes | 1-31 | \* / , – ? L W | – |
| Month | Yes | 1-12 or JAN-DEC | \* / , – | – |
| Day of week | Yes | 0-6 or SUN-SAT | \* / , – ? L # | – |
| Year | No | 1970–2099 | \* / , – | This field is not supported in standard/default implementations. |

In some uses of the CRON format there is also a *seconds* field at the beginning of the pattern. In that case, the CRON expression is a string comprising 6 or 7 fields.

**Special characters**

Support for each special character depends on specific distributions and versions of cron  
**Asterisk ( \* )**  
The asterisk indicates that the cron expression matches for all values of the field. E.g., using an asterisk in the 4th field (month) indicates every month.  
**Slash ( / )**  
Slashes describe increments of ranges. For example 3-59/15 in the 1st field (minutes) indicate the third minute of the hour and every 15 minutes thereafter. The form “\*/…” is equivalent to the form “first-last/…”, that is, an increment over the largest possible range of the field.  
**Comma ( , )**  
Commas are used to separate items of a list. For example, using “MON,WED,FRI” in the 5th field (day of week) means Mondays, Wednesdays and Fridays.  
**Hyphen ( – )**  
Hyphens define ranges. For example, 2000-2010 indicates every year between 2000 and 2010 AD, inclusive.  
**Percent ( % )**  
Percent-signs (%) in the command, unless escaped with backslash (\), are changed into newline characters, and all data after the first % are sent to the command as standard input.[[7]](https://en.wikipedia.org/wiki/Cron#cite_note-7)

**Non-Standard Characters**

The following are non-standard characters and exist only in some cron implementations, such as Quartz java scheduler.  
L – ‘L’ stands for “last”. When used in the day-of-week field, it allows you to specify constructs such as “the last Friday” (“5L”) of a given month. In the day-of-month field, it specifies the last day of the month.  
W – The ‘W’ character is allowed for the day-of-month field. This character is used to specify the weekday (Monday-Friday) nearest the given day. As an example, if you were to specify “15W” as the value for the day-of-month field, the meaning is: “the nearest weekday to the 15th of the month.” So, if the 15th is a Saturday, the trigger fires on Friday the 14th. If the 15th is a Sunday, the trigger fires on Monday the 16th. If the 15th is a Tuesday, then it fires on Tuesday the 15th. However if you specify “1W” as the value for day-of-month, and the 1st is a Saturday, the trigger fires on Monday the 3rd, as it does not ‘jump’ over the boundary of a month’s days. The ‘W’ character can be specified only when the day-of-month is a single day, not a range or list of days.  
Hash ( # )  
‘#’ is allowed for the day-of-week field, and must be followed by a number between one and five. It allows you to specify constructs such as “the second Friday” of a given month.[[8]](https://en.wikipedia.org/wiki/Cron#cite_note-8)  
Question mark ( ? )  
In some implementations, used instead of ‘\*’ for leaving either day-of-month or day-of-week blank. Other cron implementations substitute “?” with the start-up time of the cron daemon, so that ? ? \* \* \* \* would be updated to 25 8 \* \* \* \* if cron started-up on 8:25am, and would run at time every day until restarted again.