



Rate scripts

Introduction

Temp Rate Scripts provide the mechanism for converting shifts into timesheet lines when the timesheet is completed. The script must be attached to the rate scheme which is being applied.

The Commands are listed below, followed by Examples.

Scripts are case sensitive.

Rate Script Commands

comment or rem	the remainder of the line is a comment and is ignored - when using <i>comment</i> you must preserve the current indent level
foreach shift	
foreach dayslice	must be nested within foreach shift.
foreach timeslice paybandname	must be nested within foreach dayslice. In a rate scheme a pay band may have a start and end time - e.g. an Unsocial pay band may specify 1800 to 0600 as the Unsocial period which attracts a higher rate. Foreach timeslice uses the named pay band to further divide a dayslice into timeslices - those which do and do not fall within the specified time range. Use If intimeslice to determine whether the current slice is in the time range. Break treatment: Any break period is deducted from the longest slice; if slices are of equal length, then the break is deducted from the slice with the highest charge.
foreach timeslicedefault paybandname	as Foreach timeslice except that if no time range is specified for the named band, If intimeslice operates as true for the whole amount instead of false for the whole amount.
foreach timeslice hh:mm hh:mm	alternatively the start and end times may be specified literally in the script (2.18.11)
if intimeslice	used with a Foreach timeslice loop, see above. True if the slice is between the start and end time for the band.
if firsttimeslice if secondtimeslice if thirddtimeslice	Used with a Foreach timeslice loop - identifies the slices in actual time order within the dayslice.
if prevdaysliceend paybandname	executes if this is the second day slice of a shift and the time range for the payband spans midnight (or only the From time is specified).
if shift x	a shift may be of a specified type e.g. N for a night shift, and this allows the entire shift to be treated in a certain way regardless of the times or days.
if day dayname	day name may be Monday to Sunday or Holiday which looks up the Temp Holidays table. Can also use Weekday, Weekend and WeekendOrHoliday.



if previousday dayname	as above but looks at the previous day.
if nextday dayname	as above but looks at the next day.
if firstdayslice	this is the first dayslice of a shift.
if seconddayslice	this is the second dayslice of a shift.
if onedayshift	the current shift is all on one day.
if twodayshift	the current shift spans midnight.
if cancelled	the current shift was cancelled (but still flagged as payable or billable).
if bandinscheme paybandname	the named pay band exists in the current Rate Scheme.
not	any of the If conditions can be prefaced with NOT e.g. if not day Saturday.
else	if an 'else' block immediately follows an 'if' block at the same level, it will be precessed with the 'if'condition evaluates to false.
add paybandname	the current Foreach block amount (shift, dayslice or timeslice) is added to the current payband running totals. There is inbuilt protection against posting the same amount twice which simplifies the logic i.e. in the example above, hours in a shift of type N could not find their way into anything other than the Night payband.
addmin & addmax	With addable hours of x: <i>Addmax 8 MyRate</i> would: if $x < 8$ Add x else Add 8, similarly for addmin - <i>Addmin 3.5 MyRate</i> : If $x > 0$ and $x < 3.5$ Add 3.5 else Add x
post paybandname	as with Add but posts the amounts immediately as a timesheet line (i.e. not added to the summary totals).
post	posts and zeroes the summary totals.
ignore	swallows the remaining totals in the current Foreach block (i.e. prevents them from being processed by subsequent adds or posts.)
and	placed on the line after an add command (add, addupto, addmin, addmax, addlumpsum, thresholdadd) at the same indent level, returns the remaining totals to their values before that add, so that a subsequent add can re-use the same hours on a different pay band, e.g. for commission or bonus
thresholdshift paybandname1 paybandname2	if the current summary total hours for payband 2 are greater than or equal to those for payband 1 then shift the payband 1 totals into the payband 2 totals. Sometimes known as the 50% rule. Note that the separator between the pay bands is a vertical bar .
thresholdsplit thresholdhours payband1name payband2name	e.g. <i>thresholdsplit 37.5 Basic Overtime</i> would treat any Basic total hours above the threshold figure of 37.5 as Overtime hours (i.e. shift them into the Overtime totals). Note that the separator between the pay bands is a vertical bar .



thresholdadd thresholdhours payband1name payband2name	e.g thresholdadd 8 Basic Overtime would add the first 8 hours of the current Foreach block as Basic with the remainder as Overtime. Note that the separator between the pay bands is a vertical bar . You can also omit the first pay band to leave the portion below the threshold in the pot for further processing. E.g. for multi-level thresholds or a lump sum for the basic. If the threshold is greater than 24 it treats it as an aggregate threshold for the whole timesheet.
addlumpsum x paybandname	like add but exactly x hours are added, regardless of the actual hours specified in the shift (provided that they are greater than zero). Any break is ignored.
addupto time paybandname	an alternative method of time-slicing. E.g. addupto 06:00 Early. Must be used inside a foreach dayslice but NOT inside a foreach timeslice. May have a series of adduptos with increasing times to slice up the day. Each one reduces the residual amounts in the block to prevent double posting. As with timeslice, breaks are deducted from the largest slice.
showrate paybandname	the payband is added to the timesheet but with zero hours, as a place-holder for edits at completion time.

Any of the If conditions can be prefaced with NOT e.g. if not day Saturday.

Any paybandname in one of the ADD commands (add, addupto, addmin, addmax, addlumpsum, thresholdadd) may be preceded by the word PAYONLY or CHARGEONLY. This will generally be used in combination with the AND command.

Indenting

The number of leading spaces indenting each command is used to determine which commands are affected by *foreach* and *if* commands. E.g. if the leading space is removed from the final *post* command in the second example above, it will be outside the *foreach* shift loop, and the effect will be to post summary pay band totals for the timesheet as a whole (i.e. not more than one timesheet line per pay band).

Full stops, greater than signs or hyphens can be used instead of spaces when indenting commands, to improve readability.

Example

```
foreach shift
>if shift N
>>add Night
>foreach dayslice
>>if day Sunday
>>>add Weekend
>>if day Saturday
```



```
>>>add Weekend  
>>add Basic  
>post
```

Example scripts

A simple script where all time is treated added to the Basic payband until a threshold of 37 hours is reached, after which all hours are added to the Overtime payband.

```
foreach shift  
  add Basic  
thresholdsplit 37.0 Basic|Overtime  
post
```

A slightly more complex script that will step through the shifts attached to the timesheet. If a shift is of shift type N it will be posted to the Night pay band. If not it will be divided into days (i.e. two slices if it spans midnight, one if not), and the parts on Saturday or Sunday will be go to the Weekend pay band. Anything left will be go to the Basic pay band.

The totals in each band will be posted as timesheet lines for for every shift (i.e. one or more lines per shift).

```
foreach shift  
  if shift N  
    add Night  
  foreach dayslice  
    if day Sunday  
      add Weekend  
    if day Saturday  
      add Weekend  
    add Basic  
  post
```

With rates for Basic and Night - specifying the times to be treated as Night in the script rather than in the Rate Scheme

```
foreach shift  
-foreach dayslice  
--foreach timeslice 20:00 06:00  
---if intimeslice  
----add Night  
---add Basic
```



```
-post
```

With rates for Basic, Bonus and Commission - will add multiple lines for a single shift allowing for Pay Only or Charge Only lines

```
foreach shift
-add Basic
-and
-add PAYONLY Bonus
-and
-add CHARGEONLY Commission
-post
```

From:

<https://iqxusers.co.uk/iqxhelp/> - **iqx**

Permanent link:

https://iqxusers.co.uk/iqxhelp/doku.php?id=sa_ratescriptlanguage&rev=1546541568

Last update: **2019/01/03 18:52**

