



Daily Overtime Authorisation via Static Registers

SCRIPT

Overview

This script is designed to provide the facility for a customer to designate specified daily schedules as requiring overtime authorisation. Customer requirements may vary on what they want to achieve but the fundamentals remain the same. The idea is that overtime will be stored in "static registers" where they will remain in an awaiting decision state until a customer approves the overtime. At that stage the script will then move the overtime hours across to specified overtime rates. If they decline the overtime, the event is removed from the managers to-do list. The result is that employees will get paid overtime only if their manager approves it instead of automatic like the default rates do.

Dependencies

No dependencies

Operation Example

- Manager logs on to timeware
- A static register overtime approval will appear via the to-do list.
- The manager will click into this and be presented with a list of all his employees overtime awaiting a decision.
- They will either approve/decline via right clicking on them or they will double click to go into the attendance adjustments for day with the overtime
- Once approved the overtime will transition to the approved overtime rate.

Configuration

- Assigned against the daily schedules where you require the overtime approval.
- The script is split by default in to three "Bands". These are A, B & C. Further bands can be added, however please speak to timeware support about this. A band is referring to one overtime approval e.g. pre shift overtime. If in this example you wanted pre and post shift, you would utilize band A and B.
- To enable the to-do list prompt for managers, the users to-do list policy needs to be configured to show the static register awaiting decision
- Within the daily schedule "rates", you configure them in the exact same way as regular overtime except instead of paying to the approved rate i.e. rate 2, you would pay to the unapproved rate i.e. 12.

Explanation of notable fields

<p>Private Const</p> <p>BAND_A_OVERTIME_AUTHORISATION_ENABLED</p> <p>As Boolean = True</p>	<p>True or False statement. Either enables or disables the "band". Currently it is configured as True so the band is enabled.</p>
<p>Private Const</p> <p>BAND_A_OVERTIME_UNAUTHORISED_RATE</p> <p>As Integer = 12</p>	<p>Unapproved overtime rate. This is mostly used as a transitioning rate between the static register and the approved rate. Best practice will be to add 10 on to your current authorised rate e.g. if authorised rate is 2, set this as 12.</p>
<p>Private Const</p> <p>BAND_A_OVERTIME_AUTHORIZED_RATE</p> <p>As Integer = 2</p>	<p>Approved overtime rate. This is the destination after the overtime has been approved.</p>
<p>Private Const</p> <p>BAND_A_OVERTIME_AUTHORIZED_AMOUNT</p> <p>As Long = 120</p>	<p>Determines the number of hours to be paid at a certain overtime approval rate before it will then "overspill" into another rate. Useful if a customer requires e.g. 2:00 @ OT 1.3, then anything further to be paid at OT 1.5.</p>
<p>Private Const</p> <p>BAND_A_OVERTIME_AUTHORIZED_OVERSPILL_RATE</p> <p>As Integer = 3</p>	<p>The rate which is to be used as the "Overspill" rate</p>
<p>Private Const</p> <p>BAND_A_OVERTIME_AUTHORIZED_OVERSPILL_AMOUNT</p> <p>As Long = 300</p>	<p>Used to limit the amount of overtime to be paid at the overspill rate. Example here is 300 minutes i.e. 5 hours.</p>
<p>Private Const</p> <p>BAND_A_OVERTIME_STATIC_REGISTER</p> <p>As Integer = 2</p>	<p>This is the static register that you will be using. Best practice would be to go with the same number which the authorised rate will be i.e. in our examples its rate 2</p>