

Thanks Clive, that was very good!

Still having problems, unfortunately. I'm calling three APIs to make the changes. The first two appear to work OK, but the third fails with 'Grade step is invalid. Either grade step does not exist or the eff dates for the assignment are outside those for the grade step'. Sounds straightforward, but I can't find anything that looks wrong - am I missing something?

I hate to inflict reams of code on people, but here are the calls I make (in the correct order but with the surrounding code edited out). I'm only half-familiar with this stuff so am not particularly confident that what I'm doing is sensible.

VALIDATION\_MODE has been set to TRUE

ApplyDate is the date we want the changes to take effect - I've tried a few variations on the date I'm passing in, either adding or subtracting days in a bid to make it work.

UPDATE\_MODE has been set to update. Payroll want to see exceptions for future change errors etc.

```
HR_VALID_GRADE_API.create_valid_grade
    (p_validate           => VALIDATION_MODE
    ,p_grade_id           => gradeID for a given assignment
    ,p_date_from          => applyDate +1 -- +1 gives error
    ,p_effective_date     => applyDate
    ,p_position_id        => position ID for the assignment
--OUT PARAMS FROM HERE retrieved into dummy vars
    ,p_valid_grade_id     => validGradeID
    ,p_object_version_number => objectVersionNumber
    );

    hr_assignment_api.update_emp_asg_criteria
    (
    ,p_effective_date      => applyDate
    ,p_datetrack_update_mode => UPDATE_MODE
    ,p_assignment_id       => assignment ID for given assignment
    ,p_validate           => VALIDATION_MODE
    ,p_position_id        => position ID for the assignment
    ,p_people_group_id     => null value
    ,p_object_version_number => versionNo
    ,p_special_ceiling_step_id => null value
    ,p_grade_id           => gradeID for the assignment
--OUT PARAMS FROM HERE retrieved into dummy vars
    ,p_group_name          => group_name
    ,p_effective_start_date => effective_start_date
    ,p_effective_end_date  => effective_end_date
    ,p_org_now_no_manager_warning => org_now_no_manager_warning
    ,p_other_manager_warning => other_manager_warning
    ,p_spp_delete_warning  => spp_delete_warning
    ,p_entries_changed_warning => entries_changed_warning
    ,p_tax_district_changed_warning => tax_district_changed_warning
    );

    hr_sp_placement_api.create_spp
```

```
(
  p_validate          => VALIDATION_MODE
  ,p_effective_date    => applyDate + 3
  ,p_business_group_id => our BUS_GROUP_ID
  ,p_assignment_id     => assignment ID for given assignment
  ,p_step_id           => stepID for desired step - SEE BELOW
  ,p_auto_increment_flag => autoIncrementYN
  ,p_increment_number  => incrementNumber
  --OUT PARAMS FROM HEREretrieved into dummy vars
  ,p_placement_id => placementID
  ,p_object_version_number => versionNo -- assRec.versionNo
  ,p_effective_start_date => effective_start_date
  ,p_effective_end_date  => effective_end_date
);
```

This is how I'm finding the step ID used in the call above:

```
SELECT psp.spinal_point_id
FROM per_grades      pg,
     per_grade_spines_f pgs,
     per_spinal_points psp
WHERE 1=1
AND pg.name          = Our Gradename
AND applyDate BETWEEN pg.date_from AND NVL(pg.date_to,defaultDateTo)
AND pg.date_from      >= applyDate
AND pg.business_group_id = our BUS_GROUP_ID
AND pgs.grade_id       = pg.grade_id
AND applyDate BETWEEN pgs.effective_start_date AND pgs.effective_end_date
AND pgs.business_group_id = pg.business_group_id
AND pgs.parent_spine_id = psp.parent_spine_id
AND psp.spinal_point    = spinalPoint for the assignment
AND psp.business_group_id = pgs.business_group_id
```

Thanks for looking!  
Andrew