Employee Central: Managing Pay Scale Progression
The recommendations in this document are based on the functionality available up to SAP SuccessFactors release mentioned above. Future functionality can impact the recommendations provided by this document. We strive to keep these recommendations up-to-date, however, in case you find that recent new functionality has not yet been considered in the latest version of this document, please reach out to your Customer Success Manager / Partner Delivery Manager or send an email to SAPSuccessFactorsIDPDoc@sap.com.

Implementation Design Principles (IDPs) for SuccessFactors solutions are delivered by SAP for helping customers and partners on how to choose the most appropriate strategy and solution architecture for SuccessFactors implementations. IDPs are compiled taking into consideration the experience of many implementation projects and addressing frequent business requirements as well as real-life implementation challenges. They are continuously reviewed and updated as product functionality evolves. In addition, the reader is advised to read and familiarize with essential and additional product-related documentation which includes Implementation Guides, SAP Notes, SAP Knowledge Base Articles, and additional assets as referenced in this document, see chapter 8.
# TABLE OF CONTENTS

1  TERMINOLOGY .......................................................................................................................... 4

2  ABSTRACT ................................................................................................................................. 4

3  INTRODUCTION ......................................................................................................................... 4

4  BUSINESS REQUIREMENTS ....................................................................................................... 5
   4.1  FUNCTIONAL REQUIREMENTS ......................................................................................... 5
       4.1.1  Pay Scale Progression ............................................................................................... 5
       4.1.2  Compensation Update ............................................................................................... 6
       4.1.3  Exception Handling ................................................................................................. 6
       4.1.4  Reporting .................................................................................................................. 6
   4.2  TECHNICAL REQUIREMENTS ......................................................................................... 6

5  SOLUTION OVERVIEW AND CONCEPTS .............................................................................. 6
   5.1  CONFIGURATION AND DATA MODEL CHANGES .......................................................... 7
       5.1.1  Data Model Changes ............................................................................................... 7
       5.1.2  Configuration Changes ........................................................................................... 9

6  DETAILED SOLUTION .............................................................................................................. 9
   6.1  SOLUTION CONFIGURATION ....................................................................................... 9
   6.2  SAMPLE USE CASE ...................................................................................................... 14
       6.2.1  Job Run On 10.01.2018 ....................................................................................... 14
       6.2.2  Job Run On 01.01.2019 ...................................................................................... 15
   6.3  REPORTING .................................................................................................................. 16
       6.3.1  Report to List Employees Who Automatically Progressed To The Next Pay Scale Level .................................................................................................................. 16
       6.3.2  Report to list exception employees who will not be considered for automatic pay scale progression .... 17

7  ASSUMPTIONS AND EXCLUSIONS .................................................................................. 18

8  REFERENCES ............................................................................................................................ 18

9  APPENDIX ................................................................................................................................ 18
1 TERMINOLOGY

The following table explains some abbreviations used in this document.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC</td>
<td>Employee Central</td>
</tr>
<tr>
<td>ERP</td>
<td>SAP Enterprise Resource Planning often referred in the document pertains to SAP HCM on premise system</td>
</tr>
<tr>
<td>MDF</td>
<td>Meta Data Framework</td>
</tr>
<tr>
<td>RBP</td>
<td>Role Based Permissions</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>PSL</td>
<td>Pay Scale Level</td>
</tr>
<tr>
<td>PC</td>
<td>Pay Component</td>
</tr>
</tbody>
</table>

2 ABSTRACT

Grade step progression or Pay scale progression is the automatic movement of employees to the next higher pay level. This is common in industries with a unionized workforce.

This document features how Employee Central can be configured to achieve automatic grade step progression with minimal setup and maintenance. This document mainly addresses how to update the Job information employment record to reflect the new pay scale level and also provides mechanism to update the employee’s compensation information to match the pay rate of the new pay scale level.

The term grade step progression and pay scale progression is used synonymously in this document.

Refer to the IDP Employee Central: Managing Pay Scale Based Salary Increase for handling tariff changes in Employee Central.

3 INTRODUCTION

Industries that are unionized work according to agreements made in collective bargaining between the labor union and the employer(s). Collective agreements drive the way non-exempt employees get compensated based on age, seniority, time in each pay grade, number of work hours etc., It also defines the criteria of when an employee is eligible to move to a higher pay grade.

To legally comply with the collective agreement, employers need to move workforce through the pay structure as and when each employee becomes eligible based on complex calculation rules and update their compensation to reflect the pay rate as defined in the specific pay grade/pay scale level.

This document covers the usual business requirements related to pay scale progression and explains how to implement these in Employee Central, so that employee central can serve as the leading system for all the workforce’s compensation data.
4 BUSINESS REQUIREMENTS

4.1 Functional Requirements

Assumption: Pay Scale Structure is in place and defined to reflect the organizations pay structure

4.1.1 Pay Scale Progression

Pay scale progression mechanism should allow employers to automatically move employees to the next higher pay scale level based on eligibility rules governed on the basis of

1. Pay scale membership period (time spent by an employee in a given pay scale)

   Example: For the Pay Group ‘USA West Assembly Plant worker’, there are 8 pay scale levels. Refer to below table - Employee who is in Pay scale level L1 needs to wait for 9 months to become eligible to move to the next level L2, PSL L2 employee needs to wait for 12 months to become eligible to move to PSL L3, and so on and so forth.

   Once the employee reaches the highest pay scale level, he should snap to the highest pay scale level

   Pay Scale Area: USA West
   Pay Scale Type: Assembly Plant
   Pay Scale Group: USA West Assembly Plant Worker

<table>
<thead>
<tr>
<th>Pay Scale Level</th>
<th>Pay Scale Membership Period</th>
<th>Pay Rate (hourly rate in USD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA West/PLANT/ASSEMBLY/L1</td>
<td>9 Months</td>
<td>18.5</td>
</tr>
<tr>
<td>USA West/PLANT/ASSEMBLY/L2</td>
<td>12 Months</td>
<td>19.25</td>
</tr>
<tr>
<td>USA West/PLANT/ASSEMBLY/L3</td>
<td>12 Months</td>
<td>20</td>
</tr>
<tr>
<td>USA West/PLANT/ASSEMBLY/L4</td>
<td>24 Months</td>
<td>20.5</td>
</tr>
<tr>
<td>USA West/PLANT/ASSEMBLY/L5</td>
<td>24 Months</td>
<td>21</td>
</tr>
<tr>
<td>USA West/PLANT/ASSEMBLY/L6</td>
<td>36 Months</td>
<td>24</td>
</tr>
<tr>
<td>USA West/PLANT/ASSEMBLY/L7</td>
<td>36 Months</td>
<td>27</td>
</tr>
<tr>
<td>USA West/PLANT/ASSEMBLY/L8</td>
<td>NA</td>
<td>30</td>
</tr>
</tbody>
</table>

2. Actual Hours Worked by an employee

Currently this scenario is not supported within employee central as actual hours worked is not available. However, as an alternative, time data can be imported from external system such as ERP HCM or other time management system into SuccessFactors. This information can then be utilized for pay scale progression accordingly. This usecase is not covered in detail within the document, however the principle of utilizing the solution approach will be similar. The other alternative is to utilize SAP Cloud Platform Integration to read time data from the time management system, calculate the pay scale progression eligibility and update the employee records directly using APIs.

3. Employee’s Age

This is not a very common usecase however certain countries like Netherlands do use employee’s age for eligibility to progress to the next pay scale level or determine a portion of the compensation based on age. It is possible to achieve this in Employee Central, however this usecase will not be covered in this document in detail.
4.1.2 **Compensation Update**

The system should update compensation data of employees who moved to a higher pay scale level and assign the pay rate defined for the new pay scale level.

4.1.3 **Exception Handling**

Certain employees might have to be skipped from the process of automatic grade step progression, due to exception situations like

- Approvals needed in special circumstances
- Pay rate changes are overridden or postponed to a future date or marked to be reviewed again in a future date
- Employees marked for demotion etc.,

The system should provide a mechanism to exclude such employees from the automatic grade step progression.

4.1.4 **Reporting**

System should be able to generate a report listing the employees who have moved pay scale levels in a given time period and list of exception employees who are not part of the automatic pay scale progression.

4.2 **Technical Requirements**

**Maintainability:** It must not be necessary to adjust implementation/configuration on a regular basis for allowing pay scale progression. This means, no hardcoded elements within the business rules, supporting a customizable framework. Any changes to collective agreements resulting in changes to system should be limited to configuration changes and not rewriting business logic or business rules or change in business process.

5 **SOLUTION OVERVIEW AND CONCEPTS**

Automatic grade step progression can be achieved within employee central by the following 3 steps

**Step 1:** Setup pay scale structure
- Pay scale area, pay scale type, pay scale group, pay scale level
- Maintain eligibility criteria at the pay scale level

**Step 2:** Define Business rules
- Business rule to update Job information
- Business rule to update Compensation Information (Cross portlet rule associated as onSave rule of Job information)

**Step 3:** Create Off Cycle Event Batch
- Create MDF object off cycle event batch
- Assign the rule created in step 2 (rule to update job information)
- Monitor daily Bizx rules processing job

Figure 1 shows a schematic representation of the solution with relevant building blocks for configuration and administration of the process.
The eligibility criteria along with the pay rates as per the collective agreement is maintained in the pay scale structure, which is used by the business rules to progress eligible employees to the next pay scale level and subsequently update their compensation information. The off cycle event batch object and the BizX daily rules processing job provide the necessary framework to select, filter and pass employees to the business rule to perform the pay scale progression and update job information portlet. The update to job information then triggers the cross portlet onSave rule to update the compensation information portlet to reflect the pay rate associated with the new pay scale level. The admin can monitor the daily job log to check for details on which employees were progressed. Alternatively, a report can be created to list the employees that were progressed in a given time period.

5.1 Configuration and Data Model Changes

5.1.1 Data Model Changes

The Pay Scale Structure needs to be enhanced to capture the progression eligibility criteria. The following custom fields are added to the objects listed below.

- **Pay Scale Level**
  - Number of Pay Components:
    - This is based on number of pay components that are added in the pay component assignment. This is used in business rule to create/update the pay components in the compensation information portlet.
  - Membership Duration for Progression in Months:
    - This field is to capture the eligibility criteria at the Pay Scale Level. This will be used in the business rule to check if employees have met the criteria and are eligible for progression.

  Capturing the eligibility criteria at the PSL within configuration makes business rules less complex, more readable and avoids hardcoding values there by making it easier to handle changes without needing to rewrite business rules/logic. It also reduces the need to create multiple off cycle event batch objects.

  Based on the requirement, this field could have a unit other than months such as days or years. Their usage in the rules will appropriately needed to be adjusted.
**Pay Scale Pay Component**
- **Sequence Number**
  - This field will be used in business rules to select the right pay components without having to hardcode each pay component thereby minimizing changes to business rules.

**Job Information portlet changes**
- Add a field called ‘Process Auto Pay Grade Change’ and associate it with a Yes/No picklist. Populate this field with value ‘No’ for exception employees who need to be skipped from the automatic grade step progression. Ensure appropriate RBP permissions are provided to the relevant role handling the exception employees. The field will be used in the business rule to check if employee should be considered for the progression or not.

Note: Customer needs to identify as part of the business process when to set and reset this field for exception employees. For e.g., set the flag when LOA event is initiated for an employee and reset the flag after employee returns from LOA.

**Compensation information portlet changes**
- Add a field called ‘Pay Scale Relevant’. This is used to delete the existing pay scale related pay-components out of employee’s compensation before valuating the new ones coming from the next pay scale level. This is to make sure we do not modify the pay components that are not related to pay scale changes and allows for replacement of pay-components in the pay scale level configuration.
### 5.1.2 Configuration Changes

- **Event Reason**
  - Create new event reasons to identify that this is a grade step progression change.

- **Business Rules**
  - Two business rules are needed, one to update the Job Information and the other one to update the compensation information.
  - Rule to update job information – This rule will include logic to check if employee is not an exception employee and check for eligibility criteria based on employee’s current pay scale level. If the employee meets the eligibility criteria, then the job information is updated with the next pay scale level (if exists). This rule will be associated with the off cycle event batch which will be created as part of subsequent steps.
  - Rule to update compensation information – This will be an on save rule of Job Info which will trigger changes to the compensation info. The rule will update the compensation information of the employee to reflect the pay rate at the new pay scale level to which the employee progressed to.

- **Create off cycle event batch object**
  - Off cycle event batch MDF object will be used to fetch employee records based on filter criteria and pass the records to the business rule created above(rule to update job information). For the grade step progression, we will use filtering on the off cycle event batch object and also within the business rule. If the record meets all the filter/eligibility criteria, then employee is progressed to the next pay scale level if exists.

- **BizX Daily Rules Processing Batch Job**
  - This job will process all the off cycle event batch objects created and provide a detailed log.

- **Create Off Cycle Event Batch Employee Group**
  - Off cycle event batch employee groups can be created to restrict the off cycle event batch object to consider only those employees who belong to the group. Use the groups whenever possible to ensure optimal performance and throughput of the system while performing pay scale progression.

## 6 DETAILED SOLUTION

### 6.1 Solution Configuration

Let us look at applying the above solution for pay scale progression of USA West assembly plant workers, described in section 3.1.1 based on pay scale membership period

- Configure and maintain the eligibility criteria and pay rate at each pay scale level
Create an off cycle event batch employee group to include all employees from the pay scale group USA West Assembly Plant Worker

Create the business rule for updating job info

The logic for the rule

If Employee is not in exception list and membership duration for progression of current PSL is equal to time spent in current PSL and next PSL is not empty

Then

Set PSL equal to next PSL
Set job info event reason = Step Progression (PAYSPG)
Set Pay scale level entry date = Today()
In many cases, the effective date when the employee needs to move to the next pay scale level could be governed by criteria such as beginning of next pay period or 1st of next month or the following Monday etc., Such requirements can be catered to by setting the event date appropriately in the Then clause. Below screenshot shows an example where the event date is set to the beginning of the next pay period using the pay calendar.

Figure 8

If the event date is not set explicitly in the rule, the current date is used as effective date.

- Create the off cycle event batch object
  - Set base object as Job information
  - Associate the above rule and assign the employee group created
  - Set the flags:
    - Include all matched records in every run to No – See appendix for more detailed information on how this flag works
    - Include inactive records to No
    - Include only current records to Yes – This will ensure only the current record from job information is being processed. If that flag is set to No, the system will process all records both current and future dated. Appropriate logic however needs to be included in the rule to ensure correct records are being updated.
  - Setup the filters
    **Filter Field** - In our use case the eligibility for progression is driven by time spend in a PSL, we will use the ‘pay scale level entry date’ as the filter field.
    **Offset** - The filter conditions on the off cycle event batch act as an OR condition.

In the pay scale structure there are 4 distinct membership durations (9 months, 12 months, 24 months and 36 months) and hence we will create one filter each with offset values equal to membership duration, refer to Figure 9. **Operator** will be is equal to. This will ensure only those records that exactly match these durations will be passed into the rule.

An employee in PSL L4 (waiting to finish 24 months to move to L5) will pass through the filter for the first time at 9 months, second time at 12 months and a third
time at 24 months since his entry into PSL L4. However, the logic within the rule checks the membership duration of the employee against the specific eligibility defined at PSL L4 and hence will not progress the first two times but only the third time at 24 months.

E.g., Employee A was hired into PSL L4 as of 01 Jan 2017 and pay scale level entry date is set to 01 Jan 2017. A becomes eligible to move to PSL L5 on 01 Jan 2019, after being 24 months in PSL L4.

The filter on the off cycle event batch will pass the employee record to the rule
- First time on 01 Oct 2017 (9 months) using the first filter condition,
- Second time on 01 Jan 2018 (12 months) using the second filter condition and finally the
- Third time on 01 Jan 2019.

The rule will however not progress the employee into PSL L5 on the first two occasions but on the third occasion.

**Figure 9**

**Off Cycle Event Batch: SetPayScaleLevel (SetPayScaleLevel) (GradeStepProgression_USAAssemblyWrks)**

<table>
<thead>
<tr>
<th>Filter Field</th>
<th>Operator</th>
<th>Offset</th>
<th>Offset Unit</th>
<th>(1) More</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pay Scale Level Entry Date</td>
<td>is equal to</td>
<td>9</td>
<td>Month(s)</td>
<td>Details</td>
</tr>
<tr>
<td>Pay Scale Level Entry Date</td>
<td>is equal to</td>
<td>12</td>
<td>Month(s)</td>
<td>Details</td>
</tr>
<tr>
<td>Pay Scale Level Entry Date</td>
<td>is equal to</td>
<td>24</td>
<td>Month(s)</td>
<td>Details</td>
</tr>
<tr>
<td>Pay Scale Level Entry Date</td>
<td>is equal to</td>
<td>36</td>
<td>Month(s)</td>
<td>Details</td>
</tr>
</tbody>
</table>

**Additional information on setting up the filter:** As suggested above when the operator is set as “is equal to”, then the Frequency of the off cycle event batch object has to be set to ‘daily’, in order to pick up the employees to be progressed as of the exact date they become eligible for progression. Unlike in the onPrem SAP HCM system, there is no option for running the job by providing a “for period”. The system can however be configured to pick up eligible employees not just on the exact date but any day after they become eligible as well. In order to do this, use the operator value as “is on or before”. Care must be taken to ensure pay scale progression isn’t being carried out multiple times which will end up creating multiple job info records. To avoid the same, utilize the flags “Include Only Current Record” and “Include All Matched Records In Every Run” appropriately to avoid re-processing an already progressed employee. Refer to the appendix to understand the logic of how the flags work and understanding the concept of “Last Successful Run” date.

**Limitation:** Even though it is recommended to define a ‘toFilter’ for better performance, ‘toFilter’ is optional as mentioned in the implementation guide. However, when Job Information is used as the base object, not defining the ‘toFilter’ results in no records being filtered for processing by the rule. A dummy ‘toFilter’ can be setup as a workaround to the problem. Below is a sample dummy filter criteria.

<table>
<thead>
<tr>
<th>Filter Field</th>
<th>Operator</th>
<th>Offset</th>
<th>Offset Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Date</td>
<td>is on or before</td>
<td>1</td>
<td>Minute(s)</td>
</tr>
</tbody>
</table>

- Create a business rule to update compensation info

The logic for the rule

If Event reason is equal to Step Progression (PAYSPG) and PSL is not equal to previous PSL and PSL Pay component assignment amount is not null and PSL Number of PCs = 2 /* since we can have multiple number of pay components, this allows to address the PC by the seqnr index instead of having it hard-
Then

Delete all PSL relevant PCs (using field in comp pay scale relevant as yes)

Create Pay Component1 = PSL sequence 1
Create Pay Component2 = PSL sequence 2
Set job info event reason = Step Progression (PAYSPG)
Set Pay scale level entry date = Today()

Below is the snapshot of the rule

Figure 10

Update Compensation based on Pay Scale Level Change (Update_Compensation_based_on_Pay_Scale_Level_Change)

Basic Information

Start Date: 01/01/2008

Rule Type: Description

Description: Creates the necessary compensation rule when the criteria for Pay Scale progression are met such as PayScaleLevel L1 to L2.

Variables

<table>
<thead>
<tr>
<th>var.PayScaleLevelValue</th>
<th>Job Information Model.Pay Scale Level.Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>var.PayScaleLevelPreviousValue</td>
<td>Job Information Model.Pay Scale Level_Previous Value</td>
</tr>
<tr>
<td>var.EventReasonValue</td>
<td>Job Information Model.Event Reason.Value</td>
</tr>
<tr>
<td>var.EventDate</td>
<td>Job Information Model.Event Date.Value</td>
</tr>
</tbody>
</table>

If

var.PayScaleLevelValue is not equal to var.PayScaleLevelPreviousValue
and

var.EventReasonValue is equal to Step Progression (PAYSPG)

Then

Delete Job Information Model.Employment Details.Model.Compensation where Pay Scale Relevant is equal to Yes (USD)

Associate the rule on save of Job info

Figure 11
• Schedule BizX Daily Rules Processing Batch job to run

Figure 12

View Scheduled Job

View the scheduled job configuration.

* Job Name: bizxDailyRulesProcessingBatchJobType
* Job Owner: sfadmin
* Job Type: BizX Daily Rules Processing Batch

Ensure this job is run daily. The frequency of when the off cycle event batch triggers can be overwritten in individual Off Cycle Event batch objects.

Figure 13

6.2 Sample Use Case

Suppose the following three employees belong to the pay scale group USA West Assembly Plant Worker

<table>
<thead>
<tr>
<th>User ID</th>
<th>Hire Date (MM.DD.YYYY)</th>
<th>Current PSL</th>
<th>PSL Entry Date (MM.DD.YYYY)</th>
<th>Progression Eligibility Date (MM.DD.YYYY)</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>User1</td>
<td>01.01.2018</td>
<td>L1</td>
<td>01.01.2018</td>
<td>10.01.2018</td>
<td></td>
</tr>
<tr>
<td>User2</td>
<td>01.01.2018</td>
<td>L2</td>
<td>01.01.2018</td>
<td>01.01.2019</td>
<td></td>
</tr>
<tr>
<td>User3</td>
<td>01.01.2018</td>
<td>L4</td>
<td>01.01.2018</td>
<td>01.01.2020</td>
<td></td>
</tr>
</tbody>
</table>

The last column indicates the date on which each employee becomes eligible to move to the next pay scale level. This is not a field that is stored in the system but is used to illustrate the use case. The BizX daily rules processing job has been scheduled to run every day, below details show the job processing on specific days when any user(s) progress to the next level.

6.2.1 Job Run On 10.01.2018

BizX daily rules process batch job run on 10.01.2018, will process the users as below:

<table>
<thead>
<tr>
<th>User ID</th>
<th>Hire Date (MM.DD.YYYY)</th>
<th>Current PSL</th>
<th>PSL Entry Date (MM.DD.YYYY)</th>
<th>Selected by Filter</th>
<th>Modified by Rule</th>
<th>Successfully Saved</th>
<th>New PSL</th>
<th>New PSL Entry Date (MM.DD.YYYY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>User1</td>
<td>01.01.2018</td>
<td>L1</td>
<td>01.01.2018</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>L2</td>
<td>10.01.2018</td>
</tr>
<tr>
<td>User2</td>
<td>01.01.2018</td>
<td>L2</td>
<td>01.01.2018</td>
<td>NO</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>User3</td>
<td>01.01.2018</td>
<td>L4</td>
<td>01.01.2018</td>
<td>YES</td>
<td>NO</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
Resultant job log:

<table>
<thead>
<tr>
<th>Off Cycle Event Batch</th>
<th>Status</th>
<th>Number of records selected by group</th>
<th>Number of records selected by filter</th>
<th>Number of records where rule matched</th>
<th>Number of records successfully processed</th>
<th>Number of records failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>GradeStepProgression</td>
<td>Completed Successfully</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Note that all the three user's records pass the filter using the 9 months criteria, however the check within in the rule will only pass through User1 as user is in PSL L1 (with 9 months eligibility criteria).

Snapshot of Job info records of User1 before and after the job is run on 10.01.2019

Figure 14

The records would now look like below:

<table>
<thead>
<tr>
<th>User ID</th>
<th>Hire Date (MM.DD.YYYY)</th>
<th>Current PSL</th>
<th>PSL Entry Date (MM.DD.YYYY)</th>
<th>Progression Eligibility Date (MM.DD.YYYY)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>User1</td>
<td>01.01.2018</td>
<td>L1</td>
<td>01.01.2018</td>
<td>NA</td>
<td>History</td>
</tr>
<tr>
<td>User2</td>
<td>01.01.2018</td>
<td>L2</td>
<td>10.01.2018</td>
<td>10.01.2019</td>
<td>Current record</td>
</tr>
<tr>
<td>User3</td>
<td>01.01.2018</td>
<td>L4</td>
<td>01.01.2018</td>
<td>01.01.2019</td>
<td>Current record</td>
</tr>
</tbody>
</table>

6.2.2 Job Run On 01.01.2019

BizX daily rules process batch job run on 01.01.2019, will process the user’s current records as below:

<table>
<thead>
<tr>
<th>User ID</th>
<th>Hire Date (MM.DD.YYYY)</th>
<th>Current PSL</th>
<th>PSL Entry Date (MM.DD.YYYY)</th>
<th>Select by Filter</th>
<th>Modifie d by Rule</th>
<th>Successfu lly Saved</th>
<th>New PSL</th>
<th>New PSL Entry Date (MM.DD.YYYY)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>User1</td>
<td>01.01.2018</td>
<td>L2</td>
<td>10.01.2018</td>
<td>NO</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>User2</td>
<td>01.01.2018</td>
<td>L2</td>
<td>01.01.2018</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>L3</td>
<td>01.01.2019</td>
<td></td>
</tr>
<tr>
<td>User3</td>
<td>01.01.2018</td>
<td>L4</td>
<td>01.01.2018</td>
<td>YES</td>
<td>NO</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
6.3 Reporting

6.3.1 Report to List Employees Who Automatically Progressed To The Next Pay Scale Level

An adhoc report can be created to fetch the details of all employees who progressed to the next PSL using the event reason as filter criteria.

Create an adhoc report with the following:

**Domain**: Job Information (Date Range)

Define the report scope under **People**

Select desired columns relevant to the report output such as Employee Details, Pay Scale structure details, dates etc.

Complete the data type and value ID configuration

- **Filters**: Select the column Event Reason and filter value as the event reason used in business rules earlier.

Event Date can be a selection field to provide a time duration as input to run the report.
6.3.2 Report to list exception employees who will not be considered for automatic pay scale progression

An adhoc report can be created to fetch all exception employees who will be not be considered for automatic pay scale progression using the field ‘Process Auto Pay Grade Change’.

Create an adhoc report with the following:
- **Domain:** Job Information (Date Range)
- Define the report scope under **People**
- Select desired columns relevant to the report output such as Employee Details, Pay Scale structure details, dates etc.,

Complete the data type and value ID configuration

**Filters** – Select the column Process Auto Pay Grade Change and filter value as No.

Sample report output:
7 ASSUMPTIONS AND EXCLUSIONS

Business rule processing must be enabled for EC imports for off cycle event batch object to work.

8 REFERENCES

SAP Help Portal

- Initializing New Entry Date Fields in Existing JobInfo Records
- Using Batch Rules Processing using the Off Cycle Event Batch in SAP SuccessFactors
- Implementing and Configuring Employee Compensation Data in Employee Central for information on Setting up the ‘Pay Scale Structure’

9 APPENDIX

OVERVIEW OF OFF CYCLE EVENT BATCH

An Off Cycle Event Batch is a MetadataFramework (MDF) object that is used to trigger the execution of rules on a batch of records. Currently, the rules engine is capable of processing validations and making updates to attributes when a rule condition is met as a result of the transaction that is completed. However, an HRIS system has to process these rules when the condition is met at a time later than when the transaction is executed. This Off Cycle Event Batch makes it possible for some rules to be processed at the same time that the transaction is executed.

The Off Cycle Event Batch object has a filter condition and an associated rule. The filter condition will be utilized to get the records for which the condition is met and the associated rule can be used to define further filter condition (if required) and the action (SET) that needs to be taken on the record. The SET action is supported only for the HRIS employee entities (and not MDF entities) and the rule will modify only a single type of employee entity. Note that updating multiple types of entities in a single rule is not supported.

WORKING DETAILS OF THE OFF CYCLE EVENT BATCH PROCESS

The following section describes in detail how the off cycle event batch process is executed in the system, which is not covered in the implementation guide ‘Using Batch Rules Processing using the Off Cycle Event Batch in SAP SuccessFactors’. This will help understand the system generated detailed job log. It also explains with examples how certain flags work.

Off cycle event batch process will go through the following 4 stages.

Stage 1. Select Users to be processed by Off Cycle (Applicable Only if Off Cycle Event Batch User Group is configured): This is the first stage of the processor to get the user list for which off cycle process needs to be executed. If Off cycle definition contains “Off Cycle Event Batch User Group” then processor will select all users from this group (accordingly to group definition in “manage off cycle event batch groups”)

18
Stage2. Filter criteria/Filter the user records: Based on the off cycle definition, ‘tofilter’ condition and offset fields, user’s base entity records will be filtered. “Include All Matched Records In Every Run” flag: “Include All Matched Records In Every Run” works as a flag for filtering the records for off cycle event batch process and depending on the flag value off cycle processor will decide if records need to be filtered during every run or not.

<table>
<thead>
<tr>
<th>Run Seq</th>
<th>User</th>
<th>Include All matched Records in Every Run</th>
<th>Filter returns</th>
<th>Modified by Rule</th>
<th>Successfully Saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User1</td>
<td>No</td>
<td>TRUE</td>
<td>FALSE</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>User2</td>
<td>No</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>2</td>
<td>User1</td>
<td>No</td>
<td>FALSE</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>User2</td>
<td>No</td>
<td>FALSE</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

Example:
**Offcycle definition:**
Base Object: Job Information
Include All Matched Records In Every Run = No
Filter condition: Date_Filter_field = Hire_Date, Operator = “is on or before”, offset = 15, units = Years

Current design condition for "Include All Matched Records in Every Run" flag in off cycle definition:

\[
\text{Last\_successful\_run\_date} - \text{offset units} < \text{Date\_Filter\_field} && \text{Date\_Filter\_field} \text{is (Operator)}
\]

\[
\text{Current\_date} - \text{offset units}
\]

All dates below in format MM/DD/YYYY

**User job Info Records:**
User Id  Hire Date
User1  11/11/2003
User2  11/12/2003
User3  11/13/2003

**Off cycle Run 1:**
Job run details:
Last successful run date = 11-11-2018
Current date = 11/12/2018

Based on above data following condition will be generated as per current design:

\[
\text{Last\_successful\_run\_date} - \text{offset units} < \text{Date\_Filter\_field} && \text{Date\_Filter\_field} \text{is (Operator)}
\]

\[
\text{Current\_date} - \text{offset units}
\]

User1:
Offcycle Run 2:
Job run details:
Last successful run date = 11-12-2018
Current Date = 11/13/2018

Based on above data following condition will be generated as per current design:
Last_successful_run_date - offset units is < Date_Filter_field && Date_Filter_field is (Operator)
Current_date - offset units

User1:

User2:

User3:

Include Only Current Record: This field will appear only when "Job Information" is selected from "Base Object" field in off cycle event batch definition. With this field user has an option to define if off cycle needs to select only current effective record during filtering the user record or not. If "Include Only Current Record" field value is "Yes" then off cycle will select only current effective record of the user during user record filtering process.

Example Scenario:
User's Job Info Record:
Seqno User id Effective_start_date Hire_date Effective_end_date
1 111 01/01/2001 01/01/2001 01/01/2002
2 111 02/01/2001 01/01/2001 01/01/2003
3 111 02/01/2003 01/01/2001 31/12/9999

Off cycle Definition: simple off cycle event batch object without any filter
Rule: if Job Info hire_date is > 01/01/2000 then Update custom_long_field1 = "123";
a) When "Include Only Current Record" field value is set as "Yes". In above scenario off cycle process will select only current effective dated record for the user (i.e. #3) and insert only one updated record into job information table.
b) When "Include Only Current Record" field value is set as "No". In above scenario off cycle process will select all the 3 matching records and update all three records, inserts the same into job information table.

Stage3. Process Off Cycle Rule: Based on the associated rule in off cycle definition, further filtering on user records will be performed based on rule's IF condition. Then the filtered user record's info will be modified according to the rule's THEN SET condition.

Stage4. Import: This is the 'Employee Data Import' stage. In this stage records processed from stage3 will be saved/imported to Database.

Off cycle result report: The job log from the BizX daily rules processing job shows the details at each stage as depicted below in the figure. The log also provides details up to the individual user level.
### Processing Details for Off Cycle Event Batch

<table>
<thead>
<tr>
<th>Off Cycle Event Batch</th>
<th>Status</th>
<th>Number of records selected by group</th>
<th>Number of records selected by filter</th>
<th>Number of records where rule matched</th>
<th>Number of records successfully processed</th>
<th>Number of records failed</th>
<th>Run Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>GradeStepProgression</td>
<td>Completed Successfully</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0 Thu Oct 18</td>
<td></td>
</tr>
</tbody>
</table>

### Record level details

<table>
<thead>
<tr>
<th>Code</th>
<th>Selected by filter</th>
<th>Modified by Rule</th>
<th>Successfully Saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>User1</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>User2</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>