Employee Central Core Hybrid: Organization Structure Design Consideration
The purpose of this document is to recommend the design approach for creating an organizational structure in Employee Central especially when it is integrated to SAP HCM in the core hybrid model.

SAP SuccessFactors Customers: IT and HR professionals;
SAP SuccessFactors Implementation Partners: Consultants, solution architects and project managers.

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 REQUIREMENT .............................................................................................. 4
   4.1 FUNCTIONAL REQUIREMENTS ....................................................................................... 4
   4.2 PROBLEM DESCRIPTION IN DETAIL .............................................................................. 5
      4.2.1 Current Scenario With Example .............................................................................. 5
      4.2.2 Problems With The Current Process ...................................................................... 6

5 SOLUTION OVERVIEW AND CONCEPTS ........................................................................ 6
   5.1 DESIGN STRATEGY .......................................................................................................... 6
   5.2 REQUIREMENTS TO BE MET AS PART OF THE NEW DESIGN .................................. 7
   5.3 SOLUTION SECTIONS AND ITS APPLICABILITY TO CUSTOMER REQUIREMENTS ...... 8

6 DETAILED SOLUTION ....................................................................................................... 9
   6.1 BASICS OF ORG STRUCTURE ....................................................................................... 9
      6.1.1 ERP Structure .......................................................................................................... 9
      6.1.2 EC Structure ........................................................................................................... 9
   6.2 ORGANIZATION STRUCTURE AS PER THE NEW DESIGN ........................................ 10
   6.3 OPTION 1 (RECOMMENDED FOR LARGE ORGANIZATIONS WITH A BIG NUMBER OF DEPARTMENTS) ................................................................. 10
      6.3.1 Org Unit Department Object Structure and Configuration (GOM-RE-01) ................ 11
      6.3.2 Position Object Changes ........................................................................................ 16
      6.3.3 Updating Job Information (Requirement GOM-RO-03) ........................................... 19
      6.3.4 Impact on Company Structure Overview ............................................................... 20
      6.3.5 Impact on Reports (Requirement GOM-RO-04) ...................................................... 21
      6.3.6 Impact on Role Based Permission (Requirement GOM-RO-05) ............................... 21
      6.3.7 Impact on Replication and Migration .................................................................... 22
      6.3.8 Impact on Onboarding ............................................................................................ 22
      6.3.9 Impact on Recruitment ........................................................................................... 22
      6.3.10 Impact on Succession Management ...................................................................... 23
      6.3.11 Impact on LMS ...................................................................................................... 23
      6.3.12 Impact on Workflow .............................................................................................. 23
      6.3.13 Impact on Directory Search ................................................................................. 24
   6.4 OPTION 2 USE ONE FIELD OF ORGANIZATIONAL UNIT IN POSITION AND JOB INFO.............................................................. 24
      6.4.1 Impact on Reports (Requirement GOM-RO-04) ...................................................... 25
      6.4.2 Impact on RBP (Requirement GOM-RO-05) ........................................................... 25

7 ASSUMPTIONS AND EXCLUSIONS ............................................................................... 25

8 REFERENCES ..................................................................................................................... 26
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>DG- Filters</td>
<td>Dynamic Group filters</td>
</tr>
</tbody>
</table>

2 ABSTRACT

The purpose of this document is to recommend the design approach for creating an organizational structure in Employee Central especially when it is integrated to SAP HCM in the core hybrid model. This design is intended for maintaining the history of org. objects during reorganization scenarios. This document introduces the concepts of building the organization structure with one object type in Employee Central. This also introduces the concept of levels of the same object type in Job Information which helps in reporting, workflows and RBP. It also covers the impact of having such a structure in other SuccessFactors modules.

3 INTRODUCTION

Reorganizations are common in any industry. The requirement of creating, changing and moving the Org Units is part of the process. The moving of the org units would also involve moving the subordinate or child org unit along with. Apart from the child org units the position under that org unit and the employees belonging to it also must be updated with this information. In Employee Central there are multiple org related objects (BU, Division, Department), for simplicity it can be referred to as Organizational Units. During a typical reorganization, the following would have to be updated.

- Organizational unit
- All child Organizational Units
- Positions belonging to the Org Unit
- Job Information of the Employees belonging to the Org Unit.

4 BUSINESS REQUIREMENT

4.1 Functional Requirements

The following figure shows a reorganization scenario. The left sides show before the reorganization and right side shows after the reorganization.
An Organizational unit which is in level 4 (This is level from the Root Org Unit) now moves to level 3. This is just not the movement of the Department, Along the Department and the dependent objects like child departments, positions and the incumbent of the positions as well move. Notice the creation of new object types like the Division 60000001 here. Since it is moving one level above and the object types are different, new object type division is created and the old object type department is delimited.

4.2 Problem Description In Detail

For the scenario described above the following are needed to make the current record of the Department inactive and then create a Division object with the new date (Date when the Reorganization took place).

4.2.1 Current Scenario With Example

(Please refer the figure 1)

Moving Org Object (External code: 56230235) and its children from one org level 4 to another org level 3 Org level

Steps required
1. Object at level 4(department 56230235) must be made inactive on the Re-Org Date
2. Create new Organization object with the same in level 3 on the same effective date (Date of Re-Org)
3. Change the associations of the children from parent-child to Associations (The Relations between 56230235 and 57230 is a parent-child Relationship once 56230235 becomes a division with code with 60000001 this needs to be changed to associations to 57230 which continues to be a department)
4. Replication of data from EC to ERP:
5. IT1001 – Delimit existing reporting org unit relationship (56230235 reporting to 53783828 higher level Org Unit Division of EC) and created new reporting org unit relationship (60000001 reporting to 53783734) entry effective start date of Level 3
4.2.2 Problems With The Current Process

The following are problems customers encounter with the current process.

1. Though the Org unit is the same and only the relationship has changed, there is a new object created and the existing one is delimited every time (as an object type exists for each level Business Unit, Division). The historical information of this org unit is lost as a new object is created and there is no relationship to identify that this was the same object and just the relationship got changed.
2. There could be associations (not parent-child relationship) to other objects that need to be changed. This is not only the object that is moved but all the children (associated objects) must change.
3. The customers in the scenario expect that the same ID is maintained after the Re-organizations on the ERP side. So, delimiting the record on the ERP side, one would not be able to create another Org unit with the same ID, since each Org objects in Employee Central is a distinct MDF object (Business Unit, Division, Department, etc.). During replication, each object is considered as a separate entity and a new object is created during this movement. The main impact is on the downstream systems that are dependent on this information (Object IDs). Example certain Object IDs may be used as part of structural authorizations (If the children after the reorg have the same authorization then we may not have to change the profiles for them). The below table shows the place where object IDs may be used to define the Structural authorization.

<table>
<thead>
<tr>
<th>Profile No.</th>
<th>PV</th>
<th>OT</th>
<th>OID Maint.</th>
<th>Eval.path</th>
<th>Per.</th>
<th>FM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP1</td>
<td>D1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP2</td>
<td>D1</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP3</td>
<td>D1</td>
<td>O</td>
<td>100</td>
<td>ORGEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP4</td>
<td>D1</td>
<td>O</td>
<td>200</td>
<td>ORGEN</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>SP5</td>
<td>D1</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. It is also required for sometimes to use more levels in the organization hierarchy that can be used in reporting, workflows RBP, API. (Have a flattened structure with more levels required in position and Job Information).
5. This design is suitable if the organization follows purely a hierarchical tree structure.
6. Data migration from ERP to Employee central is difficult as ERP does not have Business unit and division.

If the above requirements are not relevant for the customer, this design need not be followed.

5 SOLUTION OVERVIEW AND CONCEPTS

5.1 Design Strategy

In ERP all the objects are Organizational units and are related to each other by A/B 002 relationship, the idea is to use only one object type in Employee central. The same object type is associated with each other in a hierarchical relationship.

On the EC side have only one object. Example Department and every department reports to another department through a parent-child relationship. Note that one object is recommended as it does not create new org units during re-org. If the reorganization does not happen for the first 2 levels namely the division and BU, then those objects could still be used.

The below picture illustrates the EC Structure and the ERP structure with the design mentioned above. Note the with this design of having only one object type the same object is retained without having to create new objects in EC and ERP.
The assumption of this document is that the reorganization can happen at any level. Hence the introduction of only one object type like department would help. If the reorg never happens on the Business unit level (Level 1), then you could still use the Business unit object and then have the department only structure. Also, in many customers, the positions are usually only assigned at the department level. Therefore, only the department object is replicated to the on-premise and the Business Unit and Division are not replicated but maintained as attributes in position and Job information merely as attributes to classify employees. This document focuses on a scenario where reorg can happen at any level. This is suitable for green field customers who have not yet implemented talent management modules. Since live customers might be using the employee profile standard fields like division and if the division object is not used in employee central, this would have an impact. This design would work if division is used in employee central without any relationship but used as an attribute to classify employees and not associated with the department object (Division may not be necessary to replicate to ERP).

5.2 Requirements To Be Met As Part Of The New Design

With the solution described above, we must make sure that affected entities are taken care of. The below table summarizes the requirements that should be addressed with the design of the org structure.

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirements Description</th>
<th>Detailed description and example</th>
<th>Impacted entities</th>
<th>Solution explained in section</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOM-RO-01</td>
<td>The Organizational Unit can be easily moved under a different parent</td>
<td>The Organization unit and the hierarchy below it should be considered when moving (Re-organization). The relationship with the organization unit in the current hierarchy (parent-child or association) must be changed with the movement. The Object ID should remain the same. Represent the hierarchical flattened structure in the object itself.</td>
<td>Children of the Organizational Unit, Positions and Employees</td>
<td>6.3.1</td>
</tr>
</tbody>
</table>

The table entries are structured as follows:
- **ID**: Identifies the requirement.
- **Requirements Description**: A brief description of the requirement.
- **Detailed description and example**: Provides a more detailed explanation and an example.
- **Impacted entities**: Identifies the entities affected by the requirement.
- **Solution explained in section**: References the section where the solution is explained.
5.3 Solution Sections And Its Applicability To Customer Requirements

Below table shows which ones are recommended as per the new design and which are optional (Customer-specific use cases)

<table>
<thead>
<tr>
<th>Topic</th>
<th>Optional</th>
<th>Section</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining the flattened structure as part of the Department Object</td>
<td>Yes</td>
<td>6.3.1.1</td>
<td></td>
</tr>
<tr>
<td>Level Field in the Department Object</td>
<td>Yes</td>
<td>6.3.1.4</td>
<td></td>
</tr>
<tr>
<td>Position Object changes</td>
<td>Yes</td>
<td>6.3.2.2</td>
<td>If the position does not sync to Job</td>
</tr>
</tbody>
</table>
6 DETAILED SOLUTION

Before we understand the solution, it is necessary to understand the different structures in ERP and EC.

6.1 Basics of Org Structure

6.1.1 ERP Structure

The organizational structure consists of objects and their relationships Organization Unit O, Positions S and Person P on the SAP HCM side. These objects are related by the relationship between the objects. For Org Unit to Org unit the relationship A 002/B 002 is used. For Org Unit to Positions, this is A 003 /B 003.

6.1.2 EC Structure

In EC typically, the object types involved are the Company, Business Unit, Division and Department. These are linked by associations. The object types that can be used in EC are quite flexible. The pre-delivered set of objects like the Business Unit, Division, Department can be used, in addition, MDF objects can be used to extend the Org Structure. In addition, also one to one and one to many associations can be created changed for these objects.

To align with the ERP objects, we will have to choose an object in EC. You can use either an existing object or create a new one for this. Since Department object is an object that is used quite often, we could use this instead of creating a new object. We can use the parent-child relationship between the department objects to create the hierarchy.

Custom Objects can also be used however some screens like “Directory search” and “Manage workflow” (New Home Page Version) do not have filter options for custom objects.

In this document, we choose the Department object.
6.2 Organization structure as per the new design

The figure below shows the structure in Employee central when only one object type Department is used. One department object is related to the other department object through the parent-child relationship. No associations are used.

Figure 4

6.3 Option 1 (Recommended for large organizations with a big number of departments)

The proposal is to use one org object that is department MDF object and have a parent-child relationship. Have multiple fields to represent the hierarchy in Org unit fields as custom fields on:
1. Department (optional).
2. Position (can be used to sync to Job Info)
3. Job Information

These multiple fields are also referred to as levelled structure or flattened structure. The picture below shows the sample level structure.

Figure 5
6.3.1 **Org Unit Department Object Structure and Configuration (GOM-RE-01)**

This section explains how the department object can be modelled with the flattened structure. Below is the picture of a typical department object prior to the creation of the level structure.

**Figure 6**

![Department: Production US (50150011)](Image)

The parent field is used to relate to the parent department. If you are using the Department object this then (Parent Department) field is already predefined in the object definition.

**Figure 7**

![Details](Image)

**6.3.1.1 Defining the flattened structure as part of the Department Object (Optional)**

The picture below shows how the level structure can be modelled in the department object. Define other fields like Department Level 1, Department Level 2, etc. as indicated in the figure below (This is optional see the advantages in the section below). These are custom fields defined on the department object. Level 1 is the root node, it would not have any parent. Level 2 would have level 1 as the parent. Continuing in the same way custom fields are arranged in a top-down approach would mean at every level the number of department objects shown can be filtered.
The table below shows the way the department structure before (default without enhancement) and after the changes suggested in the document.

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Start Date: 01/01/2018</td>
<td>* Start Date: 01/01/2018</td>
</tr>
<tr>
<td>* Code: 50150011</td>
<td>* Code: 50150011</td>
</tr>
<tr>
<td>Name: Production US</td>
<td>Name: Production US</td>
</tr>
<tr>
<td>Description: Production US</td>
<td>Description: Production US</td>
</tr>
<tr>
<td>* Status: Active</td>
<td>* Status: Active</td>
</tr>
<tr>
<td>Head of Department: Geoff Hill (ghill)</td>
<td>Head of Department: Geoff Hill (ghill)</td>
</tr>
<tr>
<td>Cost Center: Brothon Research &amp; Development (US13_R&amp;D004)</td>
<td>Cost Center: Brothon Research &amp; Development (US13_R&amp;D004)</td>
</tr>
</tbody>
</table>

The field criteria can be set at Level 2, Level 3 and so on with the below configuration to filter only the children of a department. Below example shows the Department Level 3 field configuration that is required to filter or show only those children belonging to the Department Level 2.

* Figure 9

Field Criteria
* Source Field Name | Destination Field Name | Default Destination Value | * Status
---|---|---|---
effectiveStartDate | effectiveStartDate | 1900-01-01 | Active
parentDepartment | cust_Department2 | | Active
6.3.1.2 Advantages of having the levels in Department Object

Please see the figure below for levels. Assume that there are 4 levels in the organization.

**Figure 10**

![Levels Diagram](image)

How would the levels in the department (flattened department structure) help? The below table gives an overview of the same.

<table>
<thead>
<tr>
<th>SL No.</th>
<th>Advantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>When you open a Department – you can easily know its hierarchy to the top from a UI point of view when a department object is opened to see the details.</td>
</tr>
<tr>
<td>2</td>
<td>The flattened structure may help in Report performance. For example, a report to get all departments below a department can be easily got by setting the query parameter like Department Level2 = “xxxx”.</td>
</tr>
<tr>
<td>3</td>
<td>These could be used in APIs to get all the children of the department. Currently, we can only go from Child to Parent. so, for example, if you want to find all the children of B, you can do a search on all departments where Level 2 is B. (see figure 10)</td>
</tr>
<tr>
<td>4</td>
<td>This could also be used to have some validations etc.</td>
</tr>
<tr>
<td>5</td>
<td>It can be used in Roles as part of the Role restrictions.</td>
</tr>
</tbody>
</table>

The disadvantage of having the level structure in the Department is having to maintain these fields when there is a reorg or a new one is created. When a new one is created the flattened structurer can be filled using a business rule that reads the level structure of the parent department.

6.3.1.3 How many Levels of the Department object is required as part of the flattened structure?

Every organization has a different level of structural depth. Some may have 4 levels, and some may have 10 levels.

The number of levels that must part of the organization structure like Department Level 1, Department Level 2, Department Level 3 and so on. There would not be a straightforward way to decide.

The following parameters may be kept in mind to come up with the optimal number.

- Will the Level be used in Reports as a query parameter?
- Will the level be used to define a Dynamic Group?
- Will this be used in the Job Information to be part of the Dynamic Roles (For Workflows).

6.3.1.4 Level Field in the Department Object (Requirement GOM-RO-06)

If the level in the org Hierarchy needs to be represented in EC, we need to have an additional Field in the Organizational object (Department) to indicate the Level (Numeric value from the Top) of the organization hierarchy.
The importance of the level is to have to use this in reporting, Business Rules, etc. we can do it in 2 ways. One way is to have a simple integer field to represent it (Level Field marked in red underlined - figure 11) Another way to do this is to define a custom object like an object Type (Level Field marked in blue underlined - figure 11) and have an integer field inside the object to indicate the level.

The “Object type” object looks like this

This field is available from 2018 Q3 in the Company Structure Overview. This is a transient field and can only be used in the Company Structure Overview currently. However, the plan is to use this field in Analytics (in the product Roadmap – Release not yet known) if the object type is used the department object would look like the picture below. We don’t need to store the level as an integer in the department object but can be stored in the “object type” object.
After the Reorg is done one may have to update the levels need to be updated. This can be done by exporting the object, correcting the data and importing it back.

**Automating level updates:**
A custom program on the cloud platform can be used to automate the process of updating the level structure in the department objects that are affected. For such a process we could use a flag to filter out the ones to be corrected. A “sync” flag is needed on the department object for this. The objective of having this field is whenever a parent department changes, the “sync” flag is updated so that the custom program would know that the underlying objects need to be updated. The “sync” flag can set through a business rule. Once the custom program has updated all the underlying or affected entities including departments, positions and job information, the flag can be updated to reflect the successful update.

For the Department where Sync is needed, it can be done through export, and once it is corrected it can be imported for mass changes. This can also be corrected for single department changes directly in the “Manage data”.

A middleware can be used to automate this process.
6.3.2 Position Object Changes

Since positions objects stores the organizational unit hierarchy as fields in the objects, it is required to have the flattened structure like the one used in the Department object. Which means you have Department Level 1, Department Level 2 as custom fields in position, etc.

Figure 16

6.3.2.1 Variations of number levels and the concept of Home Org Unit

Let us assume that we need 7 levels of Org Unit levels in the position, for example, if company A has 4 levels and company B has 7 levels. Then some of the fields (Org unit levels) in the position object will be empty. For this reason, we may need a field to store the lowest level org unit called as Home Org Unit. This could be used for HRIS sync etc.
The figure below shows the field that can be used to store the Home Department in the Position Object and subsequently in the Job information.

**Figure 18**

### Details

* Name: `cust_LowestLevelDepartms`

* Database Field Name:

  * Maximum Length: Click or focus to edit

* Data Type: `Generic Object`

Valid Values Source: `Department`

---

**6.3.2.2 Updating Positions (GOM-RE-02)**

**How are all the positions of the affected entities updated after the changes?**

The positions that are affected after the reorganization needs to be updated. The below sections describe the process of doing those updates.

**6.3.2.2.1 Updating The Position With The New Department /Org unit Hierarchy (Requirement GOM-RO-02)**

This can be done with mass changes using a CSV file. Once the CSV file is imported the Job info records of the employees are updated. This would be a manual process. The other option is to use the middleware customprogram to automate this process.
Logic would be as follows:

- All positions that are a part of the re-org is part of the position Hierarchy, either in the Department Level 1, Level 2 etc. or the home department are selected for the update.
- Update Level1 to Level n in the position (Based on the Department Hierarchy. Home department can be used for this as a basis).
- The interface will determine those successes and failed record and create a report to save on an SFTP folder. The interface will send a notification upon completion of the process in Middleware. HR admin can download the report from SFTP to correct any failed Department in SuccessFactors manually and the Middleware interface will be able to capture and reprocess it again.

6.3.2.2.2 Adapting position hierarchy affected in the Reorg (Requirement GOM-RO-02).
The position is related to each other by the parent and child relationship.

The parent position field must be updated with a new parent position. All the positions that are part of the Home Org unit being reorganized must be updated except the children(positions). The children need not be changed as their higher-level positions do not change.

The Positions that are in the triangle below are affected but only the parent positions of the highest level in the triangle, the positions under the home department must be adapted. For all the other children in the hierarchy, the same parents would be still valid.

1. Use the Mass Change Run for doing this activity

**Mass Change Run: Postion Update after Reorg (PostionUpdate1)**
Example: Rule for updating the positions in the Mass change Run.

**Figure 21**

In case the conditions are not straightforward and cannot be realized by a rule. The other option is to go for CSV upload. Download the positions for the affected Organization Unit. Change the parent position and then update the information. Mass upload can also be considered if the number of departments getting moved is many and the time taken to write a rule would be longer than the upload process.

6.3.3 **Updating Job Information (Requirement GOM-RO-03)**

Job Information would have the level structure as shown in the below picture. The level structure in job information is useful for workflows (Dynamic Role), defining permission groups and target group restrictions. The position changes are generally kept in sync with the job information. For this, the Position to Job Info sync rule must be configured. Please refer to the Implementation Guide for Position management for this. The flattened structure would also like the figure below:

**Figure 22**

Also, once you add the fields in the Job information, this can also be used in workflow Dynamic Roles to have Level Specific Approvers.
6.3.4 Impact on Company Structure Overview

The Company Structure Overview offers a hierarchical view of the organization. The root node can start from any generic object, for example, it could be a legal entity or a department. The hierarchy of departments is built using the field parent department as shown in the figure below.

There are a few points to keep in mind if the starting point is the legal entity.

- Level 1 Org units (departments) have a custom field called the legal entity
- All other departments have associations to the legal entity.

With the above design, all the departments would not get listed directly below the legal entity, only the level 1 org units will be displayed under the legal entity and further departments are displayed using the parent-child relationship of one department to the other.
6.3.5 Impact on Reports (Requirement GOM-RO-04)

The flattened structure like Department Level 1, Department Level 2 are available in Job information helps in reports such as:
"Report all employees under a department and its subsequent children departments ". This can be easily achieved.
In the absence of this, it would be difficult to create a report without repeating the departments in the query. For example, in Figure 18. In the absence of a flattened structure on the job Info, we need to repeat the departments 57230,57231,57232 for a query on all employees under the department 56230235.

6.3.6 Impact on Role Based Permission (Requirement GOM-RO-05)

For the requirement, for example, give the group access to all people that are under Level 2 Department.
You could use Fields flattened structure in the Job Information to create the groups.
Or you could use the DG-Filters to navigate from Job Information->Department ->Department Level 1 to create the group.
It is recommended to review the existing RBPs (especially the Permission Groups) for any change that must be updated based on the org changes.
It is possible to restrict the target population based on access based on some values of the Position object or Department object.

![Figure 26](image)
For the requirement, for example, give the group access to all people that are under Level 2 Department.

![Figure 27](image)
The flattened structure can also be used for defining the include, exclude criteria in Permission Groups.
6.3.7 Impact on Replication and Migration

- Only a single mapping for the standard EC object department is needed.
- The integration uses the same SAP objects before and after the configuration change. Because of the change, the external code of a department in EC could be used as the OBJID of an org unit in SAP.
- Reduced EC core and SAP org unit’s data volume for the reorg use-case.
- Additional Levels, Level field (as described in the section) must be updated part of the migration.

6.3.8 Impact on Onboarding

Standard Department object can be used in onboarding. Standard Department object can be used in Recruitment. Therefore, no Impact.

6.3.9 Impact on Recruitment

Standard Department object can be used in Recruitment. The filter options in recruitment will be based only on the home department. Filtering based on various levels within the department is not possible.
6.3.10 Impact on Succession Management

No impact as standard position management is used.

6.3.11 Impact on LMS

Use the Organization connector to build the organizational relationship, could use the FTP for the extraction and upload of the Hierarchy. Courses can be assigned at an org level and all the children of the org unit inherit from this.
One could use the parent ID relationship to import the Organizational structure. So, there is no impact with the new design.

<table>
<thead>
<tr>
<th>ORG_ID</th>
<th>ORG_DESC</th>
<th>ORG_TYP_ID</th>
<th>EMAIL_ADDR</th>
<th>ORG_ID_PAREN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Company Name</td>
<td>ENTERPRISE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>Corporate</td>
<td>DEPARTMENT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1112</td>
<td>Financial Management Services</td>
<td>DEPARTMENT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1113</td>
<td>Human Resources</td>
<td>DEPARTMENT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1114</td>
<td>Information Technology Services</td>
<td>DEPARTMENT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1115</td>
<td>Medical</td>
<td>DEPARTMENT</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1119</td>
<td>Office of the General Counsel</td>
<td>DEPARTMENT</td>
<td></td>
<td>2715</td>
</tr>
</tbody>
</table>

6.3.12 Impact on Workflow

The dynamic role would still work fine with the flattened structure. The role could be based on Department Level 1, Department Level 2 etc.
However, the flattened structure cannot be used as a filter parameter in the manage workflow UI. Only the department (home) can be used for this.
6.3.13 **Impact on Directory Search**

In the directory search, you can search based on the department field. The home Org Unit must be chosen here. However, the flattened structure cannot be used as a filter parameter UI.

6.4 **Option 2 Use One Field of Organizational Unit in Position and Job Info**

It can be used by Organizations having a lesser number of departments. In option 2 also we use only one Organization Object the Department Object, but no flattened structure like in option 1. No Department Level 1, Department Level 2, etc. In position object / Job Information you could have only one field called the department to represent the Org Unit. The lowest level department in the Organizational hierarchy is chosen and put as a field in the Position.
The side effects of this approach are the searchability of the Department as the number of departments may keep increasing as the company grows (Filtering would be a challenge). Impact on Job Information/Position will not be there as only the one field called org unit is stored and no changes to that field happen.

6.4.1 Impact on Reports (Requirement GOM-RO-04)

When a report is required to get all the employees under a department and its children. This is only possible if the departments and their children are explicitly mentioned.

6.4.2 Impact on RBP (Requirement GOM-RO-05)

If the target group criteria are based on department, the hierarchy of the department will not be considered. For example, if a person has the access to change Level 2 and below, one must add all the departments under level 2 in the target group criteria. This may also need lot of effort add each of the department manually. It is recommended to review the existing RBPs (especially the Permission Groups) for any change that must be updated based on the org changes.

7 Assumptions and Exclusions

- Business rules required to automatically calculate and assign the Org Level in Department Object during “On Save” event is not shown in the document but may be necessary.
- The possibility of updating the Levels, and the flattened structure using middleware is not documented.
8 REFERENCES

SAP Help Portal

- Implementation Handbook of Employee Central
- Employee Central Position Management