Role-Based Module Workflow in sfCodeBot

A Guide to SAP SuccessFactors Configuration Using CodeBot

Prepared by: Muhammad Arsalan Khan

Department: Software Quality Assurance

Company/Product: Net2App / CodeBot for SAP SuccessFactors

Document Type: Internal Workflow Documentation

Version: 1.0

Date: April 15, 2025

Module: Role Base Permission

Document Summary:

This document serves as a step-by-step guide to managing role-based permission configurations in SAP SuccessFactors using CodeBot. It outlines the structure, purpose, and usage of key Excel workbook sheets such as RBP Config, Groups, Roles, Permissions, and Grants, supporting automation and reverse-sync processes for efficient role and access control.

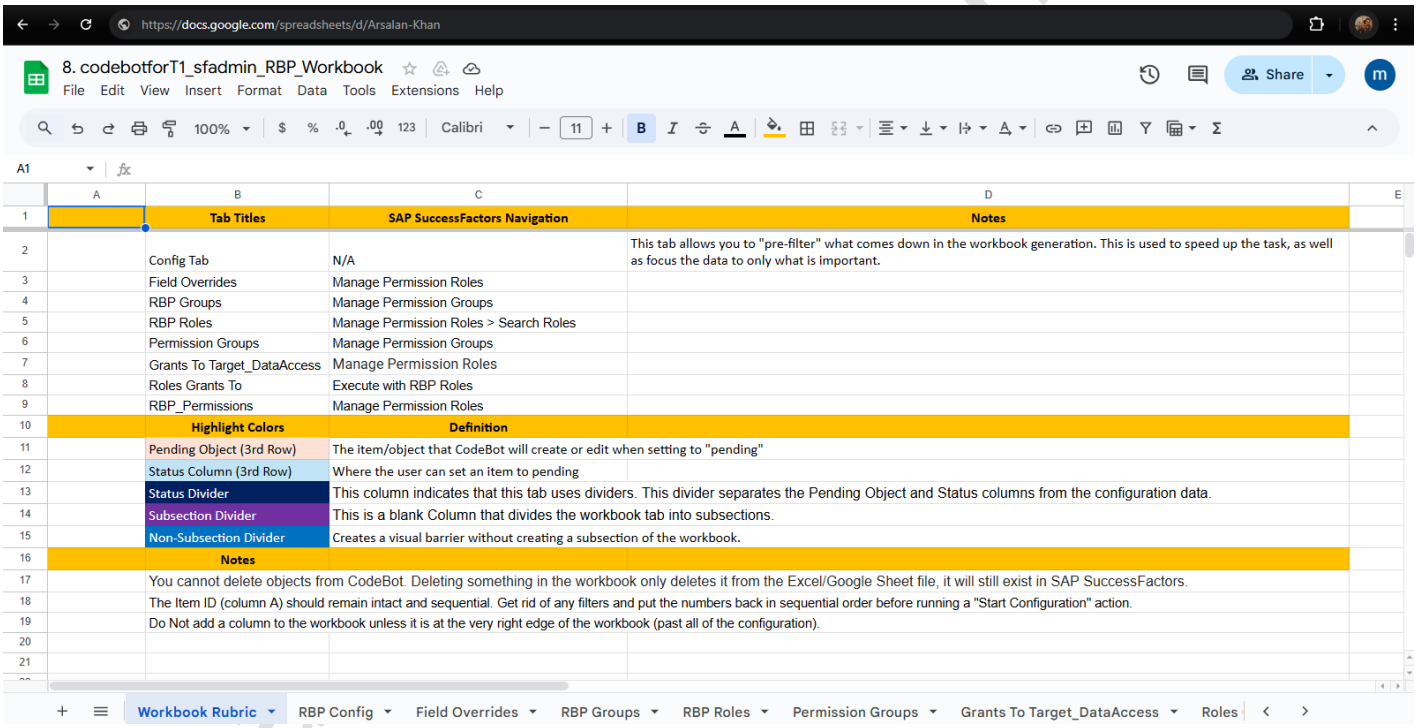
Table of Contents

1. Workbook Rubric Sheet

The Workbook Rubric Sheet provides navigation details, indicating which sheet in the workbook is linked to specific pages in SAP.

1.2 For Example:

If a Role-Based Permission (RBP) role is created through CodeBot automation, **Workbook Rubric** sheet will show where the role will be displayed in the SAP application. The user only needs to search within the workbook to identify which sheet they are working on. Next to the sheet's name, the corresponding SAP navigation will be provided. You can simply copy and paste the navigation into SAP.



https://docs.google.com/spreadsheets/d/Arsalan-Khan

8. codebotforT1_sfadmin_RBP_Workbook

File Edit View Insert Format Data Tools Extensions Help

Share

	Tab Titles	SAP SuccessFactors Navigation	Notes
2	Config Tab	N/A	This tab allows you to "pre-filter" what comes down in the workbook generation. This is used to speed up the task, as well as focus the data to only what is important.
3	Field Overrides	Manage Permission Roles	
4	RBP Groups	Manage Permission Groups	
5	RBP Roles	Manage Permission Roles > Search Roles	
6	Permission Groups	Manage Permission Groups	
7	Grants To Target_DataAccess	Manage Permission Roles	
8	Roles Grants To	Execute with RBP Roles	
9	RBP_Permissions	Manage Permission Roles	
10	**Highlight Colors**	**Definition**	
11	Pending Object (3rd Row)	The item/object that CodeBot will create or edit when setting to "pending"	
12	Status Column (3rd Row)	Where the user can set an item to pending	
13	Status Divider	This column indicates that this tab uses dividers. This divider separates the Pending Object and Status columns from the configuration data.	
14	Subsection Divider	This is a blank Column that divides the workbook tab into subsections.	
15	Non-Subsection Divider	Creates a visual barrier without creating a subsection of the workbook.	
16	**Notes**		
17	You cannot delete objects from CodeBot. Deleting something in the workbook only deletes it from the Excel/Google Sheet file, it will still exist in SAP SuccessFactors.		
18	The Item ID (column A) should remain intact and sequential. Get rid of any filters and put the numbers back in sequential order before running a "Start Configuration" action.		
19	Do Not add a column to the workbook unless it is at the very right edge of the workbook (past all of the configuration).		

Workbook Rubric | RBP Config | Field Overrides | RBP Groups | RBP Roles | Permission Groups | Grants To Target_DataAccess | Roles

2. RBP Config Sheet

2.1 Intro:

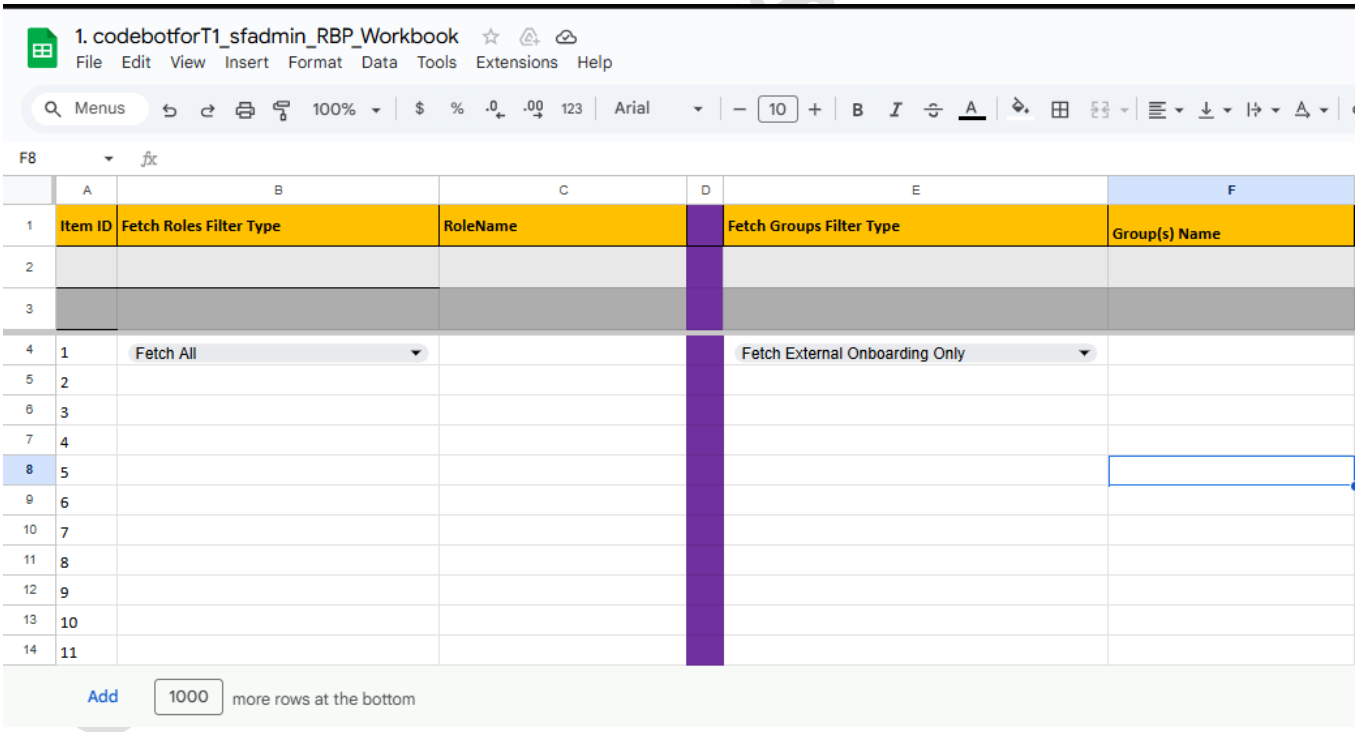
The **RBP Config Sheet** allows you to customize and reverse configurations based on your specific needs. For example, you can filter data to retrieve information for only a specific module, or limit the data to active or non-active entries, etc.

2.2 How to Run It:

To run the process, start by setting the appropriate filters in the RBP Config Sheet. This sheet contains two main filter columns:

- **Left Column:** Used to filter RBP Roles
- **Right Column:** Used to filter RBP Groups

By default, both filters are set to "Fetch All", meaning all roles and groups will be retrieved.



2.2.1 Left Column – RBP Role Filter Options:

1. **Fetch All:** Retrieves all roles.
2. **Fetch All, Except These:** Retrieves all roles *except* the ones listed.
3. **Fetch Only:** Retrieves *only* the specified role(s).
4. **Fetch Active Only:** Retrieves only roles marked as active.
5. **Fetch Inactive Only:** Retrieves only roles marked as inactive.
6. **Fetch Static Only:** Retrieves only static roles.

2.2.2 Right Column – RBP Group Filter Options:

1. **Fetch All:** Retrieves all groups.
2. **Fetch All, Except These:** Retrieves all groups *except* the ones listed.
3. **Fetch Only:** Retrieves *only* the specified group(s).
4. **Fetch Static Only:** Retrieves only static groups.
5. **Fetch Dynamic Only:** Retrieves only dynamic groups.
6. **Fetch Inactive Only:** Retrieves only inactive groups.
7. **Fetch External Onboarding Only:** Retrieves only external onboarding groups.

Once you've selected a filter from the **RBP Config Sheet**, apply it based on your requirements.

- If you chose an **RBP Role filter**, then enter one or more role names **below the "Role Name" section** in the sheet.
- If you chose an **RBP Group filter**, then enter one or more group names **below the "Group Name" section**.

You can list multiple role or group names depending on your selected filter. After setting the desired filters and input values in the RBP Config Sheet, go to **CodeBot** and Click on the **Generate Workbook** option.

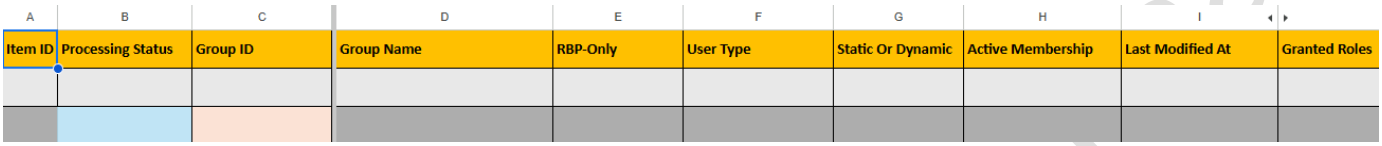
- Make sure to **select all relevant sheets** before running.
- **Important:** Always choose **“Generate Workbook”**, *not* “Configuration”.

CodeBot will process only the entries based on your configured filter and input.

3. RBP Groups Sheet:

3.1 Intro:

The RBP Groups sheet is used to manage role-based permission groups in SuccessFactors. It defines which group of users (based on filters) can be granted specific roles and permissions. This data is fetched from the system using reverse sync and only certain fields are editable.



A	B	C	D	E	F	G	H	I	
Item ID	Processing Status	Group ID	Group Name	RBP-Only	User Type	Static Or Dynamic	Active Membership	Last Modified At	Granted Roles

3.1.2 Column Breakdown:

Column Name	Description
Item ID	System-generated unique ID. Not editable.
Processing Status	Indicates the status of the entry (e.g., Processed). Not editable.
Group ID	Unique ID of the group. Not editable.
Group Name	Name of the permission group. Editable only if fetched from reverse.
RBP-Only	Indicates if this group is for RBP only. Usually “No”.
User Type	Defines user type (e.g., Employee, External Onboarding User).
Static or Dynamic	Defines if the group is dynamic (rule-based) or static.
Active Membership	Shows if users are actively assigned. Usually system-handled.
Last Modified At	Date of last modification. Not editable.
Granted Roles	List of roles granted to this group. Not editable.

⚠ Gray columns are system-controlled and must NOT be edited.

3.2 How to Use (Step-by-Step):

1. **Reverse Fetch First:**
   Start by fetching existing data using reverse sync. Only reverse-fetched records can be updated.
2. **Identify Editable Rows:**
   Locate rows with white fields (fetched entries). These are editable.
3. **Modify Fields:**
   Update the white cells like:
   - Group Name
   - User Type
   - Static or Dynamic
4. **Avoid Gray Columns:**
   Never change gray fields such as Item ID, Processing Status, Group ID, etc.
5. **Do Not Add New Rows Manually:**
   This sheet is meant only for updating reverse-synced data — do not add custom new rows.
6. **Apply Changes:**
   Once updates are made, run Configuration/Automation in CodeBot to sync changes back to the system.

Example after Change the Data In Sheet:

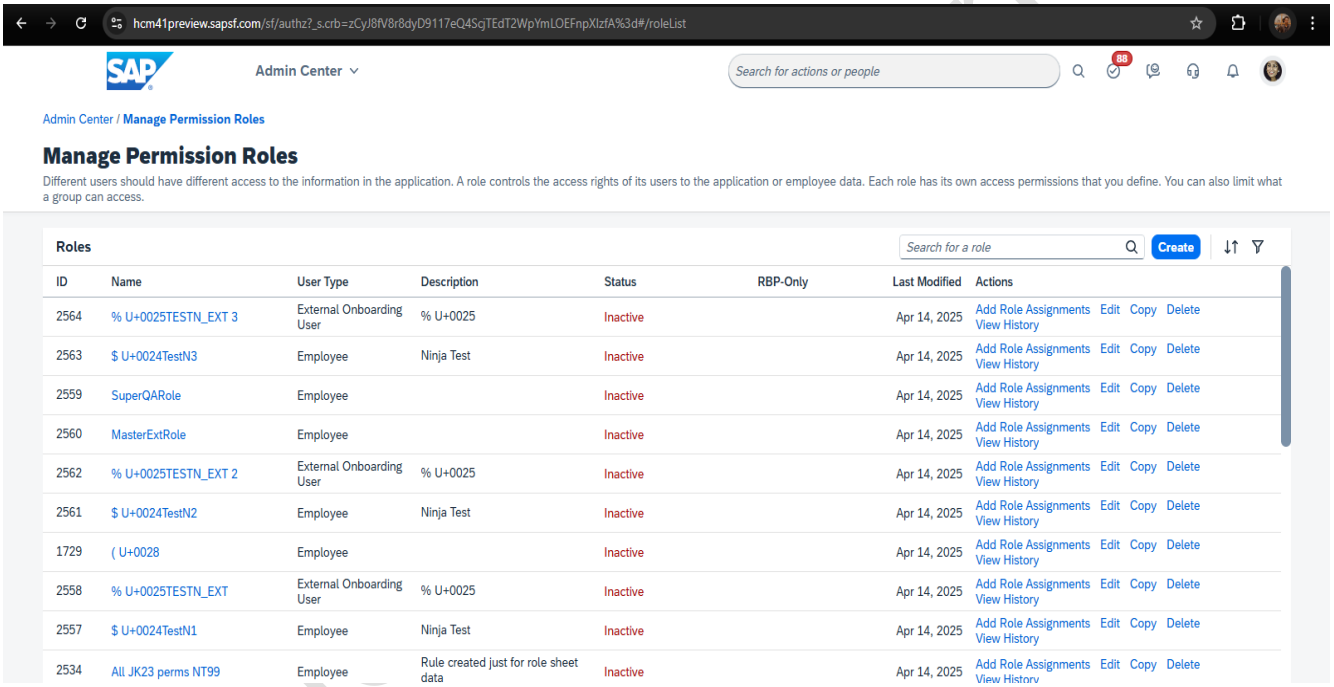
Item ID	Processing Status	Group ID	Group Name	RBP-Only	User Type	Static Or Dynamic	Active Membership	Last Modified At	Granted Roles
1	Processed	10173	Arsalan khan	No	Employee	Dynamic	0	03/05/2025	

4. RBP Roles Sheet:

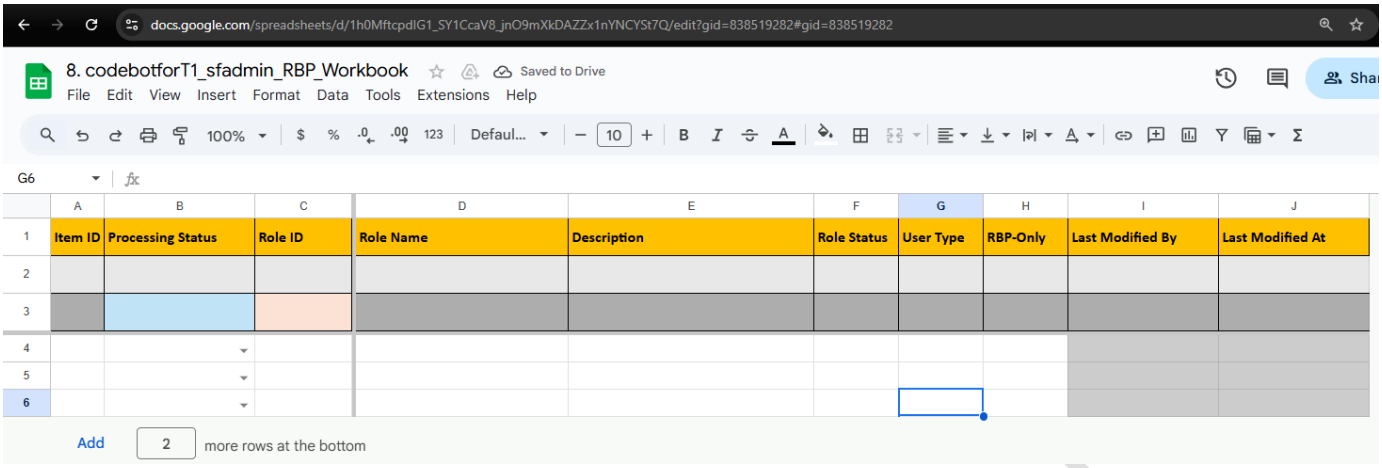
4.1 Intro:

The **RBP Roles Sheet** is used to view, create, or update Role-Based Permission (RBP) roles in SAP via CodeBot. It displays detailed information about each role, including its name, status, description, assigned users, and associated permissions. This sheet serves as a central place for managing role configurations efficiently and accurately.

Navigation on SAP: Manage Permission Roles:



4.2 Understand the Column Structure:



Each column in the sheet represents a specific attribute of an RBP role:

- **Item ID:** Sequential number for tracking each row.
- **Processing Status:** Indicates the result of processing (e.g., Processed, Error).
- **Role ID:** Unique system-generated ID for each role.
- **Role Name:** The display name of the role.
- **Description:** A brief explanation of the role's purpose.
- **Role Status:** Indicates whether the role is ACTIVE or INACTIVE.
- **User Type:** Specifies the type of user (e.g., Employee, External).
- **RBP-Only:** Shows whether this role is RBP-only (Yes/No).
- **Last Modified By:** The username of the person who last updated the role.
- **Last Modified At:** The date when the role was last modified.

Example after inserting the data in Sheet:

Item ID	Processing Status	Role ID	Role Name	Description	Role Status	User Type	RBP-Only	Last Modified By	Last Modified At
1	Processed	144	Arsalan Khan	Arsalan Khan	ACTIVE	Employee	No	sfadmin	03/12/2025

4.3 How To Use:

Step 1: Review Existing Roles

Use this sheet to review existing roles fetched from SAP. Verify key details like:

- Whether the **Role Status** is ACTIVE
- If it's an **RBP-Only** role
- Who last modified it and when

Step 2: Add or Modify Roles (If Applicable)

- You can **add a new row** to create a new role by filling in the required fields (Role Name, Description, Status, etc.).
- Or, **edit an existing row** to update role information if necessary.

Make sure to keep the Role ID field empty when creating a new role—CodeBot will auto-generate it.

Step 3: Run CodeBot to Apply Changes

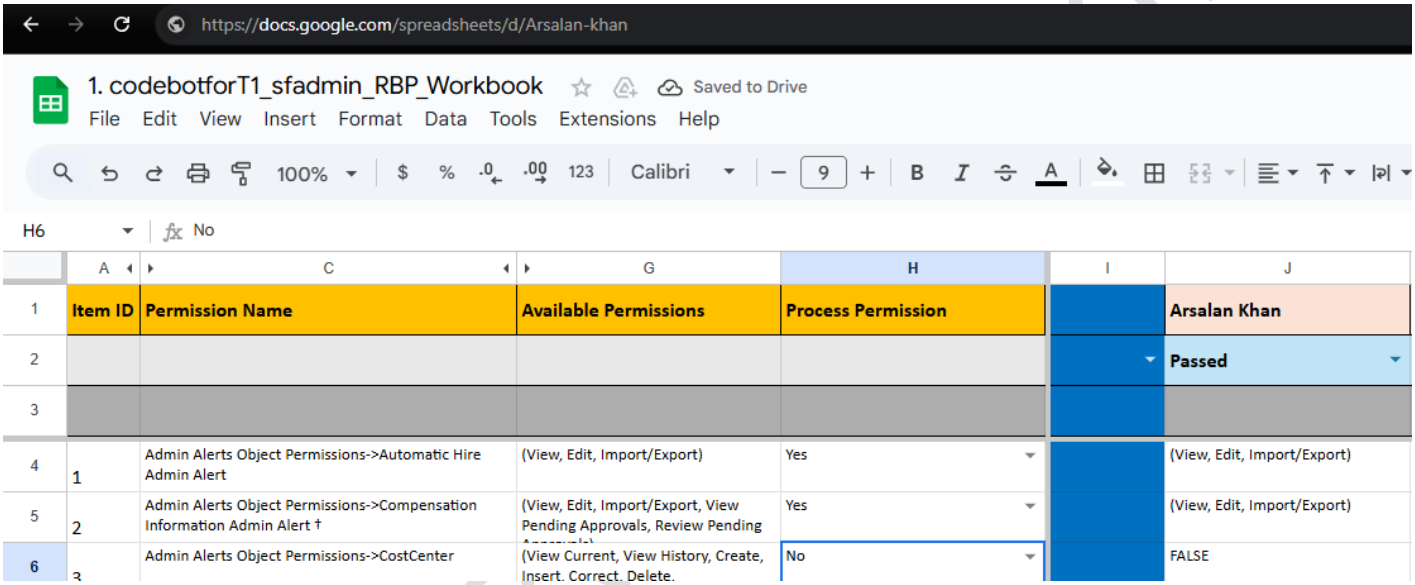
After filling in or modifying data:

- Go back to **CodeBot**
- Run the **"Automation/Configuration"** process
- CodeBot will process the sheet and update SAP accordingly

5. RBP_Permissions Sheet:

5.1 Intro:

The RBP_Permissions Sheet is used to define and manage the permissions assigned to each RBP role. This includes various modules and actions like View, Edit, Delete, and Execute. It acts as the backbone of access control, ensuring that each role has the correct level of authorization within the SAP SuccessFactors system.



	A	C	G	H	I	J
1	Item ID	Permission Name	Available Permissions	Process Permission		Arsalan Khan
2						Passed
3						
4	1	Admin Alerts Object Permissions->Automatic Hire Admin Alert	(View, Edit, Import/Export)	Yes		(View, Edit, Import/Export)
5	2	Admin Alerts Object Permissions->Compensation Information Admin Alert †	(View, Edit, Import/Export, View Pending Approvals, Review Pending	Yes		(View, Edit, Import/Export)
6	3	Admin Alerts Object Permissions->CostCenter	(View Current, View History, Create, Insert. Correct. Delete.	No		FALSE

5.2 Understand the Column Structure:

Column	Description
Item ID	A unique number to identify each permission entry in the sheet. Used for tracking.
Permission Name	The name of the specific permission or object being configured. **Example**: “Automatic Hire Admin Alert”.
Available Permissions	Lists the actions available for this permission. **Example**: View, Edit, Import/Export etc.
Process Permission	Indicates whether this permission should be processed. If it's marked "**Yes**", CodeBot will apply the permission.

5.3 How to Give Permissions Through the RBP_Permissions Sheet:

Note: This method only works if the sheet has existing data (fetched from SAP using CodeBot).

Step 1: Make Sure Data Is Fetched:

Before assigning permissions, ensure the **RBP_Permissions sheet** is already populated with permission data. You can do this by running the **"Generate Workbook"** option in CodeBot.

Step 2: Identify the Target Role

Find the **pink-colored column** in the sheet — this column is used to assign permissions to a specific role.

- In the **pink column header**, write the name of the **RBP Role** you want to assign permissions to.

Step 3: Search for the Permission

Go to the **"Permission Name"** column and find the exact permission you want to assign.

- **Example**: *"Automatic Hire Admin Alert"*

Step 4: Review Available Actions

Look at the **"Available Permissions"** column for that row to see all possible actions for that permission.

- **Example**: View, Edit, Import/Export, Delete, etc.

Step 5: Assign the Permissions

In the same row (where the permission is listed), under the pink role column:

- Type the specific actions you want to assign (e.g., **View**, **Edit**).

Step 6: Change Status to “Passed into Pending”

Set the **Status** value to **“Pending”** for that column. This marks the CodeBot for processing.

Step 7: Change the Process Permission Status to “Yes”

Set the **“Process Permission”** column value to **“Yes”** for that row. This marks the permission for processing.

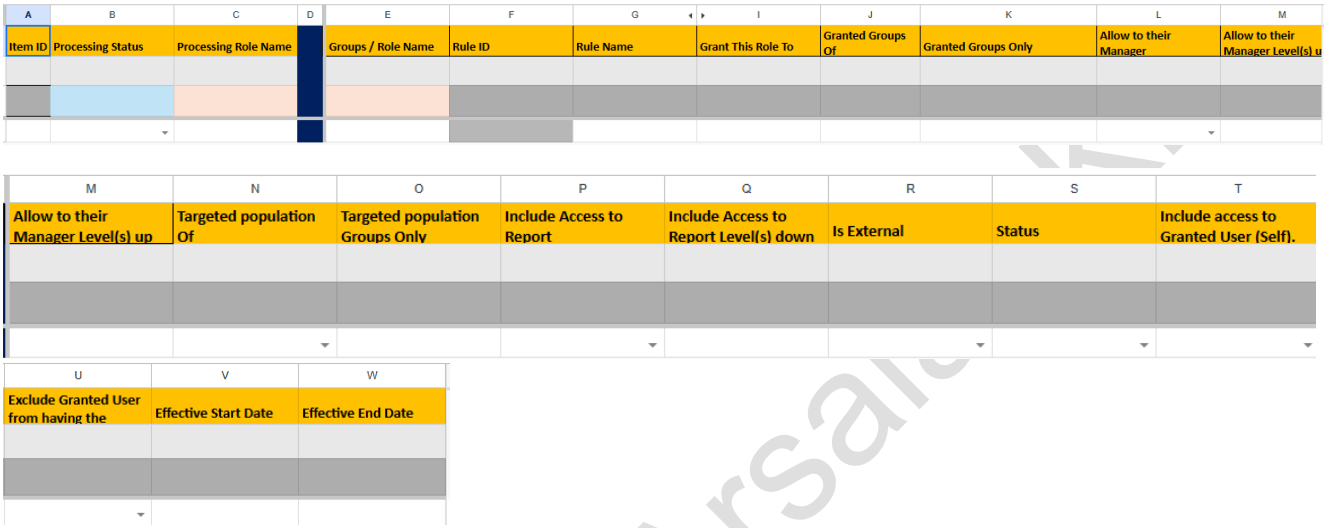
Step 8: Run the Automation

Go to CodeBot and run the **Configuration/Automation** process. CodeBot will read the sheet and update SAP with the new permissions assigned to that role.

6. Roles Grants To Sheet:

6.1 Intro:

The **"Roles Grants To"** sheet defines **which roles are assigned to which users or groups** in SAP. This sheet helps automate the **role assignment process**, eliminating the need to manually assign each role from the SAP UI.



6.2 Understand the Column Structure:

Column Name	Description
Item ID	Unique identifier for each rule.
Processing Status	Shows if the row was already processed (e.g., "Processed").
Processing Role Name	The role being processed (optional for reference).
Groups / Role Name	The role or group that is part of the assignment logic.
Rule ID	Unique ID for the assignment rule.
Rule Name	Descriptive name for the rule (e.g., “0010 Rule 1”).

Column Name	Description
Grant Role Group Pool	Reserved or internal use (optional).
Grant This Role To	Who receives the role (e.g., "Everyone", "Manager", specific groups).
Granted Groups Of / Only	Assign role based on group membership, for dynamic assignment.
Allow to Their Manager	Set to TRUE if managers of users should also get the role.
Allow to Their Manager Level(s) Up	Assign role to managers up the hierarchy (e.g., 1 level, 2 levels).
Targeted Population Of / Groups Only	Limit role visibility or usage to certain populations or groups.
Include Access to Report / Level(s) Down	Give access to data of subordinates (drill-down access).
Is External	TRUE if this assignment is meant for external users.
Status	“Active” or “Inactive”. Only Active ones are processed.
Include Access to Granted User (Self)	If TRUE, user has access to their own data under the assigned role.
Exclude Granted User from Having Access to Himself/ Herself	If TRUE, blocks self-access.
Effective Start / End Date	Define when the assignment becomes valid and when it expires.

6.3 How to Use – Step-by-Step

1. **Define Rule Information**
   Fill in the **Rule ID** You can enter any random number in the Rule ID. After executing the code, Codebot will automatically generate an ID for this rule.

   Rule Name to describe the purpose of the assignment or any name you want to give.

2. **Select Who Gets the Role**
   Use the **"Grant This Role To"** column. You can select:
   - Everyone (All Employees), Home HR Manager
   - A specific group EX: All Home HR Manager
   - [Apprentice Supervisor], etc.

3. **Control Group Scope (Optional)**
   If needed, use the **Granted Groups Of** and **Granted Groups Only** to define advanced group-based assignment rules.

4. **Manager Access (Optional)**
   - Set **Allow to Their Manager** to TRUE if the user's manager should also get the role.
   - Use **Manager Level(s) Up** to extend this to higher management.

A	B	C	D	E	F	G	I	J	K	L
Item ID	Processing Status	Processing Role Name		Groups / Role Name	Rule ID	Rule Name	Grant This Role To	Granted Groups Of	Granted Groups Only	Allow to their Manager
1	Processed	Arsalan Khan		Arsalan Khan	7794	Arsalan Khan Rule 1	Home HR Managers	All Home HR Managers	[Apprentice Supervisor]	FALSE

5. **Target Specific Populations (Optional)**
   Use **Targeted Population Of** or **Groups Only** to control visibility and role effectiveness.

   Ex: Targeted population Of : Everyone

   Ex: of Targeted population Groups Only : [Employees]

6. **Data Access Scope (Optional)**
   - Use **Include Access to Report** and **Level(s) Down** to give drill-down access.
   - This is useful for managers to view subordinate data.

   Ex: Include Access to Report : False, True
   Ex: Include Access to Report Level(s) down: All, 1, 2 etc.

7. **Configure Status and Dates**
   - Set **Status** to Active to make the rule live.
   - Define **Effective Start Date** and **End Date** if the role is time-bound.

8. **Enable Self or Restrict Self-Access**
   - Use **Include Access to Self** to allow the user to view their own data.
   - Use **Exclude from Self** to deny them that.

Item ID	Processing Status	Processing Role Name		Targeted population Of	Targeted population Groups Only	Include Access to Report	Include Access to Report Level(s) down	Is External	Status	Include access to Granted User (Self).
1	Processed	Arsalan Khan		Everyone	[Employees]	FALSE	2	FALSE	Active	FALSE

9. **Set Process Flag**
   Ensure the **Processing Status** is **Pending** for let CodeBot update it upon successful execution.

7. Grants To Target_DataAccess Sheet:

7.1 Intro:

This sheet defines which specific data sets a role has access to within SAP SuccessFactors. It controls the targeted data visibility based on object types, roles, and defined rules, enhancing data security and precision access control.

7.2 Understand the Column Structure:

Column Name	Explanation
Item ID	A unique number for each row/entry. Used for tracking.
Processing Status	Indicates the status of the automation (e.g., *Processed*, *Pending*).
Processing Role Name	The name of the role for which data access is being granted.
Groups / Role Name	Name of the group or role receiving access (can be same or linked role).
Rule ID	⚠ You must first grant the role to generate a valid Rule ID—do not enter it manually.
Grant Role Group Pool	(Optional) If the access is based on a pool or group of roles.
Exclude access to Granted User's Position	If set to TRUE, the user will not have access to their own position's data.
Target Population / Role Name	The target population to whom access applies. It can be a specific role or group.
Rule ID/Target Population	⚠ You must first grant the role to generate a valid Rule ID—do not enter it manually.

Column Name	Explanation
Target Population Object	The SAP object/module the role is being given access to (e.g., *SAP System Configuration*).
Include access to Position	If set to TRUE, the role can also access the *position-related* data of users.
Target Population Restricted Field	A specific field used to limit access (e.g., *Client ID*, *Purge Group*).
Target Population Restricted Operator	The condition used for filtering (e.g., =, in).
Target Population Restricted Value	The value or list of values used in the filter (e.g., [AK], [Arsalankhan-tech.com]).
Data Access / Role Name	The role or group that will be allowed to access the *Data Access Object*.
Rule ID/Data Access	⚠ You must first grant the role to generate a valid Rule ID—do not enter it manually.
Grant Role Group Pool / Data Access	(Optional) Role group pool assigned to data access object.
Data Access Object	The SAP data module being accessed (e.g., *Benefit Employee Profile*).
Data Access Period	Defines the time range or level of access (e.g., *Full*, *8* days/months/etc.).

7.2.1 Important Note:

You cannot manually assign a random *Rule ID*.
First, you must grant the role in the **Roles Grants To** sheet. Once the role is granted, the system will automatically generate a *Rule ID*, which you can then use in the **Grants To Target_DataAccess** sheet.

You have two ways to view your Rule ID:

1. **First**, you can open your role instance and check the Rule ID from there.
2. **Second**, you have to run Reverse on codebot for “Grants too sheet” and then check the Rule ID in that sheet.

7.3 How to Use (Step-by-Step):

1. **Define the role** under *Processing Role Name* that requires data access.
2. **Assign a Rule ID** to track this configuration (auto-generated after role grant).
3. Use *Target Population Object* and optional filters to define visibility scope.
4. Set conditions as needed:
    - Exclude the user's own position.
    - Include position access for broader visibility.
    - Apply filters using *Restricted Field*, *Operator*, and *Value* for precise control.
5. Under *Data Access Object*, select the module the role should access.
6. Specify the *Data Access Period* – either **Full** or a custom duration (e.g., **8**).
7. Once all settings are complete, run **Automation** from CodeBot to apply the configuration.

These are screenshots of all three portions after inserting the data, and they include 3 examples to make understanding easier.

Portions 1:

Item ID	Processing Status	Processing Role Name		Groups / Role Name	Rule ID	Exclude access to Granted User's
1	Processed	Arsalan khan		Arsalan khan	7802	FALSE
2	Processed	Arsalan khan Example		Arsalan khan Example	7807	FALSE
3	Processed	Arsalan khan 2		Arsalan khan 2	7808	FALSE

Portions 2:

Target Population / Role Name	Rule ID/Target Population	Target Population Object	Include access to Position	Target Population Rectricted Field	Target Population Rectricted Operator	Target Population Rectricted Value
Arsalan khan	7814	SAP System Configuration		Open Infotypes and	'=	true
Arsalan khan Example	7814	SAP System Configuration		Client ID	in	[Arsalan khan],[Arsalan
Arsalan khan 2	7814	(Do Not Use)DRTM Audit Data		Purge Group	in	[DRTM

Portions 3:

Data Access / Role Name	Rule ID/Data Access	Data Access Object	Data Access Period
Arsalan khan	7802	Benefit Employee	8
Arsalan khan Example	7802	Apprentice	Full
Arsalan khan 2	7802	Job Information	Full

8. Permission Groups Sheet

8.1 Intro:

This sheet is used to define **Dynamic Permission Groups** by applying rules based on user attributes (e.g., department, cost center, country, etc.).

Important: Only the fields and values available in your **SAP instance** will work here. Make sure you're selecting from **actual live data** in the system.

Item ID	Processing Status	Processing Group Name		Group Name	DG Pool	Field Label	Expression Value

8.2 Understand the Column Structure:

- **Processing Group Name**: The unique name for the group being created.
- **Group Name**: Often matches the Processing Group Name (acts as an identifier).
- **DG Pool**: Defines whether the group is in an **Include Pool** or **Exclude Pool** (e.g., External Onboarding User).
- **Field Label**: The user field used as a filter (e.g., Start-Compa Ratio, Date of Birth, Company, Cost Centre, etc.).
- **Expression Value**: The actual condition(s) to filter the group (e.g., specific companies, date ranges, numeric values).

8.3 How to Use (Step-by-Step):

1. **Create a new group** by assigning a name under Processing Group Name.
2. **Select DG Pool** to define whether this is an *Include* or *Exclude* group (e.g., "Include Pool 1").
3. **Pick a Field Label** from the system (e.g., *Company*, *Cost Centre*, *Date of Birth*).

   Make sure this field **exists in your SAP instance**.
4. **Define conditions** under Expression Value using valid values or ranges (e.g., [>10], [BestRun (10000)], [02/11/2024-12/11/2024]).
5. **Repeat steps 3–4** for multiple conditions for the same group if needed.
6. Once all logic is defined, **run "Configurations/Autoamtion"** from CodeBot to apply the configurations.

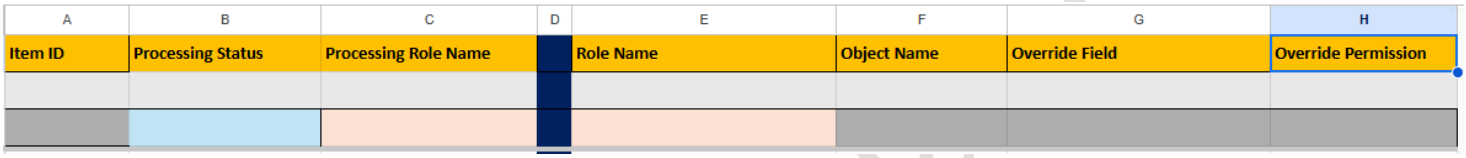
A	B	C	D	E	F	G	H
Item ID	Processing Status	Processing Group Name		Group Name	DG Pool	Field Label	Expression Value
1	Processed	Arsalan Khan Group		Arsalan Khan Group	Include Pool 1 (Exl	Start-Compa Ratio	[10],[≠11],[<4],[≤5],[>10],[≥9],[9-10],[11-10],[15-16]
2				Arsalan Khan Group	Include Pool 1 (Exl	Date of Birth	[02/11/2024],[02/11/2024-12/11/2024]
3				Arsalan Khan Group	Include Pool 2 (Exl	Job Information-Compai	[BestRun (10000)],[BestRun Bangladesh (BR_BGD)],[BestRun New Zealand (8590)]
4				Arsalan Khan Group	Include Pool 3 (Exl	Job Information-Compai	[Botswana (BWA)],[Bosnia and Herzegovina (BIH)]
5				Arsalan Khan Group	Exclude Pool 1 (Ex	Start-Compa Ratio	[10],[≠11],[<4],[≤5],[>10],[≥9],[9-10]
6				Arsalan Khan Group	Exclude Pool 2 (Ex	Start-Pay	[Argentina Monthly (AR)],[Australia Weekly (QW)]
7				Arsalan Khan Group	Exclude Pool 3 (Ex	Job Information-Cost Ce	[BRH Comp and Benefits (1710-2210)]

9. Field Overrides Sheet

9.1 Intro:

The **Field Overrides** sheet is used to customize field-level permissions for specific roles in SuccessFactors. It allows admins to override default access rights (e.g., Read Only, No Access) for selected fields under particular objects within a permission role. This is crucial when fine-tuning access to sensitive data.

9.2 Understand the Column Structure:



A	B	C	D	E	F	G	H
Item ID	Processing Status	Processing Role Name		Role Name	Object Name	Override Field	Override Permission

Column Name	Description
Item ID	Auto-generated unique identifier. **Do not edit.**
Processing Status	Shows whether the row has been processed or not. Usually auto-filled.
Processing Role Name	Internal label for the role being configured. Must match with Role Name.
Role Name	Exact name of the permission role to which override applies.
Object Name	The parent object that holds the field (e.g., Miscellaneous Permissions).
Override Field	The specific field to override (e.g., description, LegalEntity).
Override Permission	Desired permission level (e.g., No Access, Read Only, Edit).

9.3 How to Use (Step-by-Step):

1. **Choose Role Name Carefully:**
   Make sure the Role Name exactly matches the one used in Permission Roles.
2. **Identify the Object and Field:**
   Under Object Name, select the module/object (e.g., Admin Alerts Object Permissions).
   Then, specify the exact Override Field you want to control.
3. **Set Access Level:**
   Use Override Permission to assign one of the allowed values:
   - No Access – Completely restricts access.
   - Read Only – User can view but not edit.
   - Edit – Full access to view and modify (if applicable).
4. **Avoid Duplicates:**
   Do not duplicate the same Role + Object + Field combination unless necessary. It could cause conflicts.
5. **Do Not Alter System Columns:**
   Leave Item ID and Processing Status untouched unless auto-filled by the system.
6. **Apply Configuration:**
   After completing your overrides, run **Generate Workbook** from CodeBot to push changes into the system.

9.3.1 Example After Insering a Data in Sheet:

	A	B	C	D	E	F	G	H
	Item ID	Processing Status	Processing Role Name		Role Name	Object Name	Override Field	Override Permission
1		Processed	Arsalan khan		Arsalan khan	Miscellaneous Permiss	description	Read Only
2					Arsalan khan	Miscellaneous Permiss	Time Configuration for Country/Reg	No Access
3					Arsalan khan	Miscellaneous Permiss	Time Configuration for Country/Reg	Read Only
4					Arsalan khan	Miscellaneous Permiss	Maximum Additional Voluntary Cont	Read Only
5					Arsalan khan	Miscellaneous Permiss	Maximum Additional Voluntary Cont	Read Only
6					Arsalan khan	Admin Alerts Object Pe	Automatic Hire Status	No Access
7					Arsalan khan	Admin Alerts Object Pe	candidateHiredUserId	No Access
8					Arsalan khan	Admin Alerts Object Pe	cust_name1	No Access
9					Arsalan khan	Admin Alerts Object Pe	LegalEntity	No Access

9.3.1 Important:

- You can override only fields that are part of the object defined in the permission role.
- If unsure about field names, reverse-sync the role from the system first to get exact values.
- Group similar overrides together (e.g., all CostCenter fields) for easier readability.