Great 👍 Let’s break this lesson down in **simple terms first** and then I’ll give you the **short summary points at the end**.

## 📌 **Person and Employment Objects in Employee Central**

### 1. ****What this chapter is about****

1. Employee Central doesn’t just hold company data (foundation objects like location, division, legal entity).
2. It also stores **employee-specific data** → things about the person (personal info) and their work (employment info).
3. These are called **HR Objects**.

They are split into:

1. **Person Objects** → who the person is.
2. **Employment Objects** → what job they do.

Both are configured in **Succession Data Model (SDM)** and **Country-Specific Succession Data Model (CSF SDM)**. Admin tool used: **Manage Business Configuration (BCUI)** in Admin Center.

### 2. ****Features of HR Data****

1. **Effective-dated** → keeps history of changes (address changes, promotions, salary updates).
2. **Connects** with other SAP SuccessFactors modules (like Payroll, Compensation).
3. **Reportable & exportable** → can be pulled into reports.
4. **Secured** → controlled by Role-Based Permissions (RBP).
5. **Configurable** → standard fields + country-specific fields + custom fields.

### 3. ****Person Objects (Personal Information)****

These are predefined elements (HRIS elements) that store data about an individual.

Examples:

1. **nationalIdCard** → National IDs (support CSF, workflow, but not effective-dated).
2. **homeAddress** → Employee’s address (effective-dated, supports CSF & workflow).
3. **personalInfo** → Name, DOB, etc. (effective-dated).
4. **personInfo** → Biographical info (not effective-dated).
5. **phoneInfo, emailInfo, imInfo** → Contact details (not effective-dated).
6. **personRelationshipInfo** → Dependents (effective-dated).
7. **workPermitInfo** → Visa/work permit info.

👉 Important: Some are effective-dated, some are not. This affects whether history is stored.

### 4. ****Employment Objects (Work Information)****

These are predefined elements that store job-related data.

Examples:

1. **jobInfo** → Job/Org info (effective-dated).
2. **jobRelationsInfo** → Manager/mentor/supervisor relationship (effective-dated, no CSF).
3. **employmentInfo** → Contract details like hire date, probation, etc. (not effective-dated).
4. **compInfo** → Compensation (effective-dated).
5. **payComponentRecurring** → Recurring payments like monthly salary (effective-dated).
6. **payComponentNonRecurring** → One-time payments like bonuses (not effective-dated).

👉 Some add-ons (Global Assignment, Pension Payout) add more objects but not core.

### 5. ****Effective Dating****

1. **Effective-dated objects** keep history → Old record closes, new record opens.
2. Example:
3. Employee lives at 214 E 6th Ave from 2011 to 2019.
4. Moves to 315 W 10th Ave in 2020.
5. System stores BOTH addresses with start/end dates.
6. History panel shows old crossed out, new highlighted.
7. **Non-effective dated objects** → No history, just overwrite.
8. Example: personInfo (biographical info).

### 6. ****Field Types in HRIS****

1. **custom-string** → free text.
2. **custom-long** → integer.
3. **custom-double** → decimal.
4. **custom-date** → date field.

Used when adding custom fields to personal or employment info.

## ✅ Summary (Key Takeaways)

1. **HR Data = Person Data + Employment Data**.
2. Stored in **Succession Data Model (SDM)** and **CSF SDM**, managed via **BCUI**.
3. **Person objects** → personal details (address, IDs, contacts, dependents).
4. **Employment objects** → work/job details (job info, pay, compensation).
5. **Effective-dated objects** keep history of changes (e.g., address, job info).
6. **Non-effective dated objects** overwrite old data (e.g., personInfo).
7. You can extend HR data with **custom fields** of type string, long, double, date.

Would you like me to make a **comparison table (Person vs Employment objects)** with effective-dated / CSF support / workflow support so you can revise it faster before exams?

# **Introducing Hire and Termination Screens**

Perfect 👌 this is a big chunk, so let me first **explain it to you step by step in simple terms**, then I’ll give you the **summary with the most important points at the end**.

## 🟢 Hire and Termination Screens in Employee Central

### 1. ****Hire Process****

1. **Who can do it?** → Usually HR admins.
2. **How it’s done?** → Through **Add New Employee (New Hire) wizard**.
3. This is a guided form to enter all required info about a new employee.
4. Templates can be **customized**:
5. Example: A template for interns (fewer details) vs. one for permanent employees (more details).
6. If you don’t set up custom templates, the system uses the **standard template**.

👉 **Real-life example**: Think of it like when you open an account in a bank. They have a “New Customer” form. Depending on whether you’re opening a savings account, salary account, or business account, they might rearrange what details they ask. That’s what SuccessFactors does with hire templates.

### 2. ****Employee Identifiers (IDs)****

SAP SuccessFactors uses **three main IDs** to keep track of employees:

|  |
| --- |
| ID What it’s for Example |

|  |  |  |
| --- | --- | --- |
| **User-ID** | Tied to **employment data** (job info, compensation, history). Each job record has a unique one. | Employee can have multiple jobs → multiple User-IDs. |
| **Person-id External** | Tied to **personal data** (name, DOB, email, address). Always **one per person**. | If same person has 2 jobs, both link back to this one person-id. |
| **Username** | Used to **log in**. Can also be multiple (if they hold different employments). | jdoe123 for one job, jdoe456 for another. |

👉 **Key rule**:

1. **User-ID** → created at every new hire.
2. **Person-id-external** → created once per person.
3. **Username** → created at hire, can be changed.

### 3. ****Duplicate Hire Validation****

When you try to hire someone, the system checks:

1. Name + DOB
2. Name
3. National ID

If a match is found, you can:

1. **Accept Match** → Keep old personal info, add new employment info.
2. **Rehire with new employment** → Start fresh employment record but keep person link.
3. **Ignore Matches** → Treat them as a completely new person.

👉 Example: Dennis Jackson worked before, left, now coming back.

1. Accept Match → old personal info carries forward.
2. Rehire → he gets a new job record but tied to the same person-id.

### 4. ****Role-Based Permissions (RBP) for Hire/Rehire****

1. Hiring activities are controlled under **Manage Hires** permission section.
2. This ensures only authorized people can hire or rehire.

### 5. ****Manage Pending Hires****

1. One place to **finalize hires** initiated from:
2. Employee Central
3. Recruiting module
4. Onboarding module
5. Shows pending approvals, status, and issues.
6. If automatic hire is set up (from Recruiting/Onboarding), the system can add the employee automatically (no HR manual work).

### 6. ****Termination Process****

1. When someone leaves → HR opens their **profile** → chooses **Terminate**.
2. Must enter:
3. Termination Date
4. Termination Reason
5. “OK to Rehire” option (important for future rehire decisions).

👉 If the person is a **manager**, the system asks: Who should the direct reports now report to? Options:

1. Upper manager
2. Another manager (you choose)
3. Individually assign each report to a new manager

⚠️ **Rule**: Only one termination record can exist per employee.

### 7. ****Report No-Shows****

1. Situation: An employee is hired, but never shows up to work.
2. Instead of terminating (which can mess up payroll), use **Report No-Show**.
3. Effect:
4. Their record becomes **inactive immediately**.
5. No active work date is created → avoids payroll errors.
6. Available only within **30 days after hire date**.

👉 Example: You hired John Doe to start Monday. He never comes. Instead of saying “Terminate” (which assumes he worked and left), you mark him as **No-Show**.

### 🟡 Practical Task Examples (from your snippet)

1. Hire Michael Vorm with a New Hire template.
2. Rehire Dennis Jackson (system detects duplicate).
3. Terminate Larry Lindsley (early retirement).

## 📌 Summary – Most Important Points

1. **Hire screen = customizable wizard** (different templates for different employee types).
2. **Identifiers**:
3. User-ID = job/employment
4. Person-id-external = personal identity (only one per person)
5. Username = login ID
6. **Duplicate Check** prevents accidental double hires. Options: Accept, Rehire, or Ignore.
7. **Manage Pending Hires** = one-stop place to track and complete hires from multiple sources.
8. **Termination screen** = requires reason, date, and handling direct reports.
9. **Report No-Show** = avoids payroll issues if someone never actually joined.
10. Permissions are controlled through **RBP → Manage Hires**.

👉 Would you like me to now make **possible exam-style questions + answers** from this topic (like the ones you asked earlier) so you can practice?

# **Importing New Employees into SAP SuccessFactors Employee Central**

Alright 👍 let’s break this down step by step so you clearly understand how **Importing Employees into Employee Central** works.

## 🟢 Importing New Employees into SAP SuccessFactors EC

### 1. ****Ways to Add Employees****

Employees can be added in many ways:

1. **Add New Employee (New Hire) wizard** → manual hire by HR
2. **Manage Pending Hires** → pull from Recruiting/Onboarding
3. **FTP/Connectors** → system integrations (for large-scale loads)
4. **Import Employee Data tool** → uploading CSV files (focus of this chapter)

👉 This chapter = how to use **Import Employee Data** (Action Search).

### 2. ****Business Keys****

1. Every record in Employee Central is identified by a **Business Key** → a unique combination of fields.
2. **If record is effective-dated** → business key includes **person\_id\_external/user\_id + start date + record type**.
3. **If record is not effective-dated** → only needs **person\_id\_external** (like biographical info).

👉 Example:

1. **Address** record → (person\_id\_external + start date + address type).
2. **Biographical Info** (non-dated) → (person\_id\_external only).

📌 Always include business keys in your import files.

### 3. ****Permissions for Import****

1. Controlled by **Role-Based Permissions (RBP)**.
2. Steps:
3. Enable **Employee Central Import Settings → Import Employee Data**.
4. Grant access to specific entities (Job Info, Personal Info, etc.).

🔑 Extra Import Permission Options:

1. **Import Employee Data** → lets user access the tool.
2. **Enable workflows for imports** → workflows get triggered on import.
3. **Enable business rules** → rules execute when data is imported.
4. **NO\_OVERWRITE rules** → even fields marked no-overwrite can be updated.
5. **Forward Propagation** → effective-dated imports push data forward to future records.

👉 Important: **Basic User Import** (very common) can be further restricted via RBP.

### 4. ****Import Process****

1. You import data by preparing **CSV templates**.
2. Import is usually done in a **specific order** (dependencies):
3. **Basic Import** (creates the user record)
4. **Biographical Info** (DOB, gender, etc.)
5. **Employment Details**
6. **Job Information** (position, manager, department)
7. **Compensation Info** (salary/pay)
8. **Personal Information** (address, email, phone, etc.)

👉 If you skip the order, system may fail because required links (like User ID) don’t exist yet.

### 5. ****Tips to Avoid Import Failure****

1. Always use a **fresh template** (download from Import tool).
2. Fill **all required fields** (like UserID, person\_id\_external, hire date).
3. Use **codes/IDs** (e.g., ENG for job code) not names (like “Engineer”).
4. Make sure picklists/objects used in fields exist in the system.
5. For effective-dated records:
6. Don’t manually mess with end dates unless you know hierarchy.
7. Keep all effective-dated records in **one CSV** per entity.

### 6. ****Exercise Example****

Ace Corp adds 3 employees (Jean, Stephen, John).

1. Prepare 6 CSVs with provided data (like job code = ENG, company = ACE\_USA).
2. Upload in the order: **Basic → Bio → Employment → Job → Compensation → Personal**.
3. Validate: search for them in People Search after upload.

## 📌 Summary – Key Takeaways

1. **Import Employee Data tool** = upload CSVs to create/update users.
2. **Business Keys** uniquely identify records → always required in import.
3. Import permissions controlled via **RBP** (Employee Central Import Settings).
4. **Import order matters**: Basic → Biographical → Employment → Job → Compensation → Personal.
5. Use **fresh templates** to match latest system configuration.
6. Always use IDs/codes instead of names when importing.
7. Import failures often come from missing required fields, misconfigured picklists, or wrong end dates.

👉 Do you want me to create a **mini Q&A set (like your earlier 4 questions)** from this Import lesson so you can quiz yourself?

# **Updating Employee Information in SAP SuccessFactors Employee Central via Import**

Perfect! 👍 Let’s break this lesson into **exam-friendly points** so you can revise quickly.

## ****Updating Employee Information via Import (Employee Central)****

### 1. ****Centralized Services****

1. Umbrella term for services controlling data processes in Employee Central.
2. Ensures consistency in:
3. **Importing Employee Data** (via Import Employee Data page or OData API).
4. **Saving on History UI** (clock icon).
5. **Saving on Editing UI** (pencil icon).
6. **Enabled by default** for all entities, except:
7. Dependents (enable via Company System and Logo Settings).
8. Add New Employee, Add New Employee for Fixed Term, and Rehire Inactive Employee.

### 2. ****Importing Updates****

1. Updates done via **Import Employee Data** tool.
2. **Choose Load Type**:
3. **Full Purge** → Deletes old data and replaces with new (not recommended for small fixes).
4. **Incremental Load** → Adds/updates only changed data.

### 3. ****Partial Import****

1. Special type of incremental load.
2. Use **&&NO\_OVERWRITE&&** to keep existing field values unchanged.
3. Empty fields (without &&NO\_OVERWRITE&&) → become **null**.
4. **Not supported for entities**:
5. Work Permit Info
6. Job Relationships
7. Dynamic Role
8. Workflow

### 4. ****Deleting Employee Data via Import****

1. Deletion possible in **Incremental Load** with operation type:
2. **DELETE** → Removes full record (e.g., Compensation Info, Personal Info, Job History, Email, Phone, etc.).
3. **DELIMIT** → End-dates a record (e.g., Addresses, Global Info, Job Relationships, Person Relationships, Recurring Pay).
4. For MDF (Metadata Framework) objects:
5. **Delete** → Parent object.
6. **Delimit** → Specific child record.
7. **Clear** → All child records from parent.

### 5. ****Exercise Example****

1. Error correction in **PWelshJobhistoryimport.csv**:
2. Remove “Contract Type” → not required field.
3. Insert record via **Job Information History**:
4. Effective Date: **03/01/2012**
5. Event: **Job Reclassification**
6. Event Reason: **Job Reclassification**
7. Job Classification: **Vice President, Operations (VP-OPS)**

### ****Key Takeaways****

1. Centralized Services ensures consistent processing of data.
2. **Incremental Load** preferred for updates (not Full Purge).
3. **Partial Import** allows selective updates with &&NO\_OVERWRITE&&.
4. DELETE vs DELIMIT determines whether a record is removed or end-dated.

👉 Do you want me to now **make a quick Q&A quiz** (like your earlier request) from this lesson so you can check if you’ve memorized the points?

# **Applying Mass Changes in SAP SuccessFactors Employee Central**

Got it 👍 Let’s structure this lesson into **clear exam-ready notes** for you.

## ****Applying Mass Changes in Employee Central****

### 1. ****Mass Data Change Tools****

There are **two main UI-based tools** for bulk updates in Employee Central:

1. **Manage Mass Changes**
2. **Mass Data Management**

### 2. ****Manage Mass Changes****

1. Used by **system admins** to apply **uniform updates** to multiple employees at once.
2. Supports **Job Information** and **Job Relationships**.
3. Example: Relocating all San Francisco employees to San Mateo in one action.
4. **Steps**:
5. Provide **job name**.
6. Create a **group of impacted employees** (like RBP group).
7. Enter **effective date**.
8. Select if change is for **Job Info** or **Job Relationship**.
9. Choose the **field** and enter **value**.
10. Select **Event Reason** (only for Job Info).
11. Save and initiate the job.

### 3. ****Mass Data Management****

1. More **advanced and flexible UI tool** for bulk changes.
2. **Scope**: Job Information & Position Management.
3. **Key Features**:
4. **Advanced filtering**: build precise target groups.
5. **Change review**: draft → highlight modifications → share with up to 5 colleagues.
6. **Audit & reporting**: download logs.
7. **Field customization**: up to **50 Job Info fields** can be set as filters or editable.
8. Supports **custom fields**.
9. **Steps**:
10. Grant permissions:
11. Enable Mass Data Management
12. Miscellaneous → Mass Change Job / Mass Changes Job Configuration
13. Create Mass Data Change UI in **Manage Data**.
14. Use **Job Information category** to create request.
15. Apply **filters** → select employees (up to 50).
16. Modify fields → apply changes.
17. Save draft (optional: share).
18. Submit → triggers all **onSave & postSave rules** (can disable via Manage Business Configuration).

### 4. ****Summary (Exam Pointers)****

1. **Manage Mass Changes** → supports **Job Information + Job Relationships**.
2. **Mass Data Management** → supports **Job Information + Position Management**.
3. Mass Data Management provides **more flexibility** (filters, drafts, custom fields, logs).
4. Both tools avoid the complexity of traditional imports.

⚡ Would you like me to now create a **side-by-side comparison table** (Manage Mass Changes vs Mass Data Management) so you can instantly differentiate them in exams?

# **Configuring the Employee Central Succession Data Model with Business Configuration UI (BCUI)**

Perfect — this chapter is heavier, but let’s break it into **step-by-step digestible notes** so you understand exactly what’s going on before we move to summary points.

# ****Configuring the Employee Central Succession Data Model with BCUI****

### 1. ****Succession Data Model (SDM)****

1. In **Employee Central (EC)**, employee data is controlled by **data models (XML files)**.
2. Two main XML models:
3. **Succession Data Model (SDM):** defines **global HRIS elements** (employee data structures like personal info, job info, etc.).
4. **Country-Specific Succession Data Model (CSF-SDM):** defines **country-specific fields** (like National ID for India, Social Security Number for US).
5. Normally, consultants had to edit XML files → now we use **Business Configuration UI (BCUI)** to avoid manual XML editing.

### 2. ****BCUI (Business Configuration UI)****

1. A **UI tool** for admins/consultants to configure HRIS elements (instead of touching XML).
2. Requires the Succession + Country-Specific Data Models to be uploaded **once** during implementation.
3. Benefits:
4. No XML editing.
5. Direct updates in the system.
6. You can validate changes with the **Check Tool** (recommended after upload).

⚠️ Limitation:

1. **No undo / backup.** Every change is permanent.
2. By default → **view mode**. To edit → Take Action → Make Correction.

### 3. ****What can be done in BCUI?****

1. **For Employee Central (HRIS elements):**
2. Set label/default label.
3. Enable/disable elements or fields.
4. Assign **business rules** to elements.
5. Configure **field properties** (field type, required, visible, editable, etc.).
6. **For Employee Profile (Talent modules):** configure user info fields.

💡 Example: In Biographical Information, you can enable an **attachment field** so employees can upload documents.

### 4. ****Person Types in BCUI****

1. **Person Type = subgroup of people** (e.g., Employee, Candidate, Dependent, Onboardee, Contingent Worker).
2. Each Person Type can have its own set of fields.
3. Supported only for HRIS elements (not for custom XML editing).
4. When you create a new Person Type:
5. Fields are copied from the base element.
6. You can configure attributes: **enabled, mandatory, visibility**.
7. Availability of person types depends on enabled modules (e.g., Dependents Management → Dependent type appears).

### 5. ****Dynamic Group Filters (DG-Filters)****

1. Used when defining **permission groups (RBP People Pools)**.
2. Example: You want a group of employees filtered by Location or Job Level.
3. BCUI allows you to add HRIS fields as **filters** for building these groups.

**Steps to add a DG-Filter:**

1. Go to **Manage Business Configuration** → Filters → Dynamic Group Filters.
2. Select an HRIS element (e.g., Location).
3. Select HRIS field (e.g., Location ID).
4. Save.
5. Field becomes available as filter in **People Pools** for RBP.

### 6. ****Using Generic Objects in Succession Data Model****

1. Sometimes companies want **custom foundation objects** (beyond standard ones like Legal Entity, Department, Location).
2. Example: **Parking spots per location**.
3. Generic Objects can be created in MDF (MetaData Framework) → then linked in Succession Data Model via BCUI.

**Steps to assign Generic Object to SDM:**

1. Go to **Manage Business Configuration** → select HRIS element (e.g., Job Information).
2. Add new **custom field**.
3. Configure it:
4. Type of Reference Object = Foundation Object.
5. Reference Object = Generic Object ID (e.g., cust\_parking).
6. Destination Field Value + Source Field Name (for filtering associations).
7. Save + assign permissions.

💡 Example: Employee’s Job Info → Location = Arlington → Parking field auto-filters only garages tied to Arlington.

# ****Summary (Exam-Ready Points)****

1. **BCUI** = UI tool for editing Succession + CSF Succession Data Models (instead of XML).
2. **Person Types** allow you to configure specific fields for groups like Employee, Candidate, Dependent, etc.
3. **Dynamic Group Filters (DG-Filters)** let you add Employee Central fields for filtering users in RBP groups.
4. **Generic Objects** can be linked to HRIS elements as custom fields (e.g., Parking → filtered by Location).
5. ⚠️ BCUI has **no undo** and always requires careful saving.

👉 Do you want me to also make a **diagram-style mapping** (like a flow: XML → BCUI → Person Types / DG-Filters / Generic Objects) so you can visualize how these all connect?

# **Configuring the Employee Central Succession Data Model using XML**

Perfect 👍 You’ve pasted the section about **Configuring the Employee Central Succession Data Model using XML**. Let me first explain it in a very simple and detailed way (like I’m teaching you from scratch), and then I’ll give you a **summary with most important points** at the end.

## 🔹 What is this about?

In SAP SuccessFactors Employee Central, **employee data** (like job info, personal info, supervisor details, etc.) is stored and structured using something called **Succession Data Model (SDM)**.

There are **two ways** to configure it:

1. **BCUI (Business Configuration UI)** → A graphical UI in the system (easy, but limited).
2. **XML** → More technical, but powerful, gives version control and flexibility.

This chapter is about **using XML**.

## 🔹 Why use XML when BCUI exists?

1. **Bulk/Simultaneous changes**: XML makes it easier to edit multiple global & country-specific fields at once. BCUI often throws errors.
2. **Version control**: Every time you upload an XML in Provisioning, a **backup version** is saved. In BCUI, no history is kept.
3. **Complex configurations**: Some advanced customizations are easier in XML.

## 🔹 Steps to configure using XML

### 1. ****Export the Succession Data Model (SDM)****

1. Go to Admin Center → Export Succession Data Model.
2. Download the current XML file.
3. Always work on the latest version.
4. Rename with a new version number so you don’t overwrite old ones.
5. Up to 100 past versions are stored automatically.

### 2. ****Edit in XML editor****

1. Open the file in an XML editor (like Notepad++ or Oxygen XML).
2. Find the **element** (like personInfo, jobInfo).
3. Inside it, find or add **fields** (hris-field).

Example:

1. They want to **rename "Supervisor Level" → "Management Level"**.
2. In XML:

<hris-field id="supervisor-level" visibility="both">

<label>Management Level</label>

</hris-field>

1. They also want to add a **new field** Additional Job Details.

<hris-field id="custom-string8" visibility="both">

<label>Additional Job Details</label>

</hris-field>

⚠️ Important: Make sure you add this field **inside the correct element (jobInfo)**, not in the wrong block.

### 3. ****Upload back into system****

1. Go to Provisioning → Import/Export Data Model.
2. Upload the new XML file.
3. Add a comment (so you know what change was made).
4. Submit.

### 4. ****Set Permissions****

Even if a field exists, users can’t see it unless you give permission.

1. Go to **Manage Permission Roles**.
2. Choose the role (e.g., System Admin).
3. Search for new fields (Management Level, Additional Job Details).
4. Give all permissions (View, Edit, Delete, Correct, etc.).

### 5. ****Test it****

1. Go to an employee profile (e.g., Marcus Hoff).
2. Check **Employment Information**.
3. You should now see:
4. Management Level field (instead of Supervisor Level).
5. Additional Job Details field (custom).

## 🔹 Summary (Key Points to Remember)

1. **Succession Data Model (SDM)** → XML file that defines employee data structure.
2. **Why XML?**
3. Easier for bulk changes.
4. Backup version history (up to 100).
5. **Process**:
6. Export current XML → Edit → Upload in Provisioning.
7. Add/modify fields using <hris-field>.
8. Give permissions in Manage Permission Roles.
9. Test in employee profile.
10. Example:
11. Renamed “Supervisor Level” → “Management Level”.
12. Added new field → “Additional Job Details”.

👉 So in short: **BCUI = easy UI but no history**. **XML = powerful, allows version backup, good for bulk edits**.

Would you like me to also **compare side by side (BCUI vs XML)** in a simple table for exam/interview readiness?

# **Configuring the country and region-specific fields for Succession Data Model in Employee Central**

Great 👍 You’ve pasted the section on **Configuring the Country and Region-Specific Fields for Succession Data Model**. Let’s break it down step by step in a beginner-friendly way, then I’ll give you the **summary with most important points**.

## 🔹 What is this about?

Every country has different HR requirements. For example:

1. **Addresses**: US has “State”, UK has “County”, India has “PIN code”.
2. **National IDs**: US → Social Security Number (SSN), France → INSEE, India → Aadhaar.
3. **Job Info & Employment Info**: Some countries need extra fields like “Travel Distance” (Germany), “Probation” (USA), “Hiring Initiative” (France).

This is handled in **Country-Specific Succession Data Model (CSF SDM)**. So instead of one common global structure, you can have **different fields for each country**.

## 🔹 Which elements support country-specific fields?

Not all HRIS elements need country-specific versions. The main ones are:

1. **globalInfo** → Global Information (ethnicity, veteran status, religion, etc.)
2. **homeAddress** → Address formats (State, County, Province, etc.)
3. **nationalIdCard** → National IDs with specific formats per country.
4. **jobInfo** → Job Information (travel distance, EEO data, etc.)
5. **compInfo** → Compensation Info.
6. **employmentInfo** → Employment Details.

## 🔹 Examples explained

### 1. ****National ID****

1. Preconfigured for >100 countries.
2. Defined by:
3. **Country ID** → Which country.
4. **Format ID** → Type of ID (like SSN, PAN, INSEE).
5. **Regex** → Ensures correct format.
6. Can store **multiple IDs over time** with start & end dates. (e.g., Old Passport → Expired, New Passport → Active).

### 2. ****Home Address****

1. Example:
2. UK → “County” (from picklist COUNTY\_GBR).
3. Canada → “Province” (from picklist PROVINCE\_CAN).
4. Preconfigured, but you can also add extra fields.

### 3. ****Global Information****

1. Appears inside the **Personal Information block** in People Profile.
2. Example: For USA → Ethnic Group, Veteran Status (with picklists).
3. Example Exercise: Ace Corp added **Ethnicity** and **Religion** for Germany.

### 4. ****Job Information****

1. Varies by country based on **Legal Entity**.
2. Example: Germany needs **Travel Distance**.
3. US might require **EEO fields**.

### 5. ****Employment Information & Compensation Info****

1. No pre-delivered fields → You must create custom ones.
2. Example:
3. US → “Probation Waived” (Picklist Yes/No), “End of Probation” (Date).
4. France → “Hiring Initiative” (Text), “Previous Service From” (Date).

## 🔹 How do you configure these fields?

### Method 1: ****Manage Business Configuration (BCUI)****

1. Go to Manage Business Configuration.
2. Expand the HRIS element (like globalInfo, jobInfo).
3. Add country-specific version (globalInfo\_DEU, employmentInfo\_USA).
4. Add or enable fields (custom-string1, custom-date1, etc.).
5. Set: Label, Enabled, Visibility, Type (picklist, text, date).

### Method 2: ****CSF Succession Data Model XML****

1. More advanced → edit the XML directly.
2. But BCUI is preferred for most cases.

### Always: ****Set Permissions****

1. After creating a new field → go to Manage Permission Roles.
2. Example: DEU-Travel Distance, USA-Probation Waived, FRA-Hiring Initiative.
3. Give View/Edit/Delete permissions.

### Test in Employee Profiles

1. German employee → Sees “Travel Distance”.
2. US employee → Sees “Probation Waived” & “End of Probation”.
3. French employee → Sees “Hiring Initiative” & “Previous Service From”.
4. Others → Don’t see these fields.

## 🔹 Summary (Key Takeaways)

1. **CSF Succession Data Model** allows configuring **country-specific HR fields** (address, IDs, job, employment info).
2. **Pre-delivered fields exist** for >100 countries (e.g., SSN in US, INSEE in France, State/Province in addresses).
3. **Custom fields** can be added using **BCUI or XML**.
4. **National ID** uses Country ID, Format ID, and Regex to ensure correct input.
5. **Global Info** (e.g., Ethnicity, Religion) sits inside Personal Information.
6. **Job Info, Employment Info, Comp Info** → often customized per country.
7. **Permissions are mandatory**: Without role-based permissions, new fields won’t appear.
8. **End result**: Employees see fields relevant only to their country.

Would you like me to also create a **side-by-side comparison of Global Succession Data Model vs Country-Specific Succession Data Model (SDM vs CSF SDM)** for better clarity?

# **Configuring HRIS Synchronization in SAP SuccessFactors Employee Central**

Perfect 👌 You’ve shared the lesson on **Configuring HRIS Synchronization**. Let’s go step by step in detail (beginner-friendly), and then I’ll give you the crisp summary points.

# 🔹 What is HRIS Synchronization?

1. **HRIS = Human Resource Information System**.
2. **HRIS Sync = One-way synchronization** of employee data **from Employee Central (EC) → User Data Tables & Employee Profile**.
3. Why? Because some modules (Performance, Goals, Compensation, Recruiting) still rely on older tables called **User Data Tables (UDF)**.

👉 Example: You update an employee’s department in **Job Information (EC)** → HRIS Sync copies it to the **Employee Profile/User Tables** → Talent modules can use it.

⚠️ Warning: Never update UDF/User Tables directly with Basic Import → it causes **data inconsistency**.

# 🔹 Why is HRIS Sync Important?

1. **Employee Central** = Core HR system (personal info, job info, employment info).
2. **Employee Profile** = Needed for **Talent processes** (performance reviews, compensation cycles, route maps, reports).
3. Talent modules can’t always pull directly from EC → they rely on synced UDF data.
4. So HRIS Sync ensures **both EC & Talent modules stay consistent**.

# 🔹 How HRIS Sync Works

1. **Background job** (HRIS Sync Job) runs on a schedule.
2. Looks for changes in EC → Updates UDF & Employee Profile.
3. **Triggers automatically** when:
4. Change made in UI (current or past-dated).
5. Data imported into EC.
6. A scheduled HRIS Sync Job runs.

Types of Jobs:

1. **Full Sync** → All data.
2. **Incremental Sync** → Only changes since last successful sync.
3. Jobs can be **One-Time** or **Recurring**.

# 🔹 HRIS Sync Mappings

How does the system know which EC field goes to which UDF field? → **Mappings**.

### 1. ****Hard-Coded Mappings**** (default, can’t be changed)

Examples:

1. employmentInfo.start-date → hireDate
2. jobInfo.emplStatus → status
3. jobInfo.manager-id → managerId
4. jobInfo.job-code → jobCode
5. jobInfo.department → department
6. jobInfo.division → division
7. jobInfo.location → location
8. jobRelationsInfo.rel-user-id → manager types (HR, Matrix, Second, Custom)

⚠️ If you try to create a custom mapping for these, system blocks it.

### 2. ****Custom Mappings**** (created by consultant/admin)

1. Done either in **Succession Data Model XML** (<hris-sync-mappings>)
2. Or in **BCUI (Manage Business Configuration → HRIS Sync Mappings)**.

Example: ACE Corp wants to map **Standard Weekly Hours (Job Info)** → Employee Profile custom field.

Steps:

1. Go to BCUI → find custom07 standard field → rename label to Standard Hours.
2. Go to BCUI → HRIS Sync Mappings → Add (+).
3. HRIS Element: Job Info
4. Field: Standard Weekly Hours
5. Target Field Type: Standard Field
6. Target Field: custom07 (Standard Hours)
7. Save.
8. Go to Configure People Profile → Add Standard Hours to Employee Information block.
9. Test: Change Standard Hours in Job Info → Save → Check Employee Profile → Value updated.

# 🔹 HRIS Sync Jobs

Created in either:

1. **Provisioning** (backend, usually consultants), or
2. **Scheduled Job Manager** (Admin Center).

Example: If someone uploaded wrong UDF (Basic Import) → Data mismatch → Run a **One-Time Full Sync Job** to correct.

Job fields:

1. Job Name: HRIS Sync Custom
2. Job Type: Sync HRIS Data
3. Sync Type: Since last sync OR Full Sync
4. Occurrence: One-Time / Recurring
5. Owner: Person responsible
6. Notifications: Send emails when job runs
7. Run job manually → Check status in Job Monitor.

# 🔹 Key Takeaways (Summary)

1. HRIS Sync = **One-way sync** → EC → UDF & Employee Profile.
2. Needed because **Talent Modules rely on UDF data**, not directly on EC.
3. **Triggered** by UI changes, imports, or scheduled jobs.
4. **Job Types**: Full Sync / Incremental Sync; One-Time / Recurring.
5. **Hard-coded mappings** (e.g., hireDate, jobCode, managerId, department) exist by default.
6. **Custom mappings** possible via BCUI or XML (e.g., Standard Hours → custom07).
7. Jobs managed in **Scheduled Job Manager** or **Provisioning**.
8. ⚠️ Never update User Data File manually if EC is enabled → leads to inconsistencies.

👉 Do you want me to also create a **flow diagram (step-by-step visual)** of how HRIS Sync works (EC → UDF → Talent Modules), so you can revise faster before exams?