

# HR350

## **Programming in HR**

THE BEST-RUN BUSINESSES RUN SAP



© SAP AG 2003

- SAP R/3 Enterprise
- 2003/Q3
- 50063940



### Requirements:

- **HR100** (*Essentials of Personnel Administration*)
- **HR120** (*Essentials of Personnel Planning*)
- **BC400** (*ABAP Workbench: Foundation*)
- **BC405** (*Techniques of List Processing*)
- **Experience of programming in ABAP**



### Target group:

- IT staff
- IT administrators

### Duration:

- 5 days



© SAP AG 2003

### Note:

- The training materials do **not** constitute a **teach-yourself program**. They **complement the course instructor's explanations**. Your material includes space for making note of additional information.
- There may not be enough time to complete all exercises during the course. The exercises contain additional examples that are dealt with during the course. Participants can use these examples to consolidate their knowledge once they have completed the course.



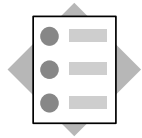
### **Contents:**

- **Course Goals**
- **Course Objectives**
- **Course Content**
- **Main Business Scenario**



**This course will prepare you to:**

- **Program your own HR reports using logical databases and HR-specific statements**
- **Create customer-specific infotypes**



**At the conclusion of this course, you will be able to:**

- **Use the HR logical databases and the Join and Projection views**
- **Use HR-specific statements**
- **Import and process payroll results**
- **Create customer-specific infotypes**

## Preface

Unit 1	<b>Course Overview</b>	Unit 7	<b>Cluster Database Tables in HR</b>
Unit 2	<b>Data Structures in Personnel Administration</b>	Unit 8	<b>Data Structures in Personnel Planning</b>
Unit 3	<b>The Logical Databases PNP/PNPCE</b>	Unit 9	<b>The Logical Database PCH</b>
Unit 4	<b>Join and Projection</b>	Unit 10	<b>Customer-Specific Infotypes</b>
Unit 5	<b>Specific Commands</b>	Unit 11	<b>HR ABAP Features</b>
Unit 6	<b>Repetitive Structures and List Display</b>		<b>Appendix</b>
			<b>Introduction to HR</b>

© SAP AG 2003

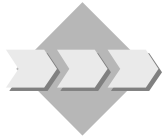
### ■ Note:

In the sample source code contained in these training materials, all ABAP keywords and the names of HR data structures are printed in upper case.

In the solutions for the programming exercises, only the ABAP keywords are written in upper case.

- One part of the appendix contains additional overheads, which the trainer can incorporate into the corresponding units, where appropriate.

Another part contains material for an introduction to HR, which the trainer may use at the start of the course, if necessary.

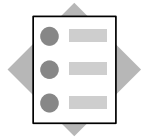


- **All components of the HR system have been implemented at your company.**
- **The Personnel Department in your company requires its own reports and customer-specific infotypes to cover special requirements.**



## Contents

- Infotypes
- Technical data structures of infotypes
- Check and control tables



**At the conclusion of this unit, you will be able to:**

- **Describe the data structures of Personnel Administration**
- **Use the check and control tables of HR infotypes as a source of information**



Course Overview

Cluster Database Tables in HR

 **Data Structures in Personnel Administration**

Data Structures in Personnel Planning

The Logical Databases PNP/PNPCE

The Logical Database PCH

Join and Projection

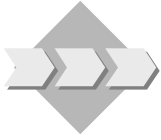
Customer-Specific Infotypes

Specific Commands

HR ABAP Features

Repetitive Structures and List Display

© SAP AG 2003

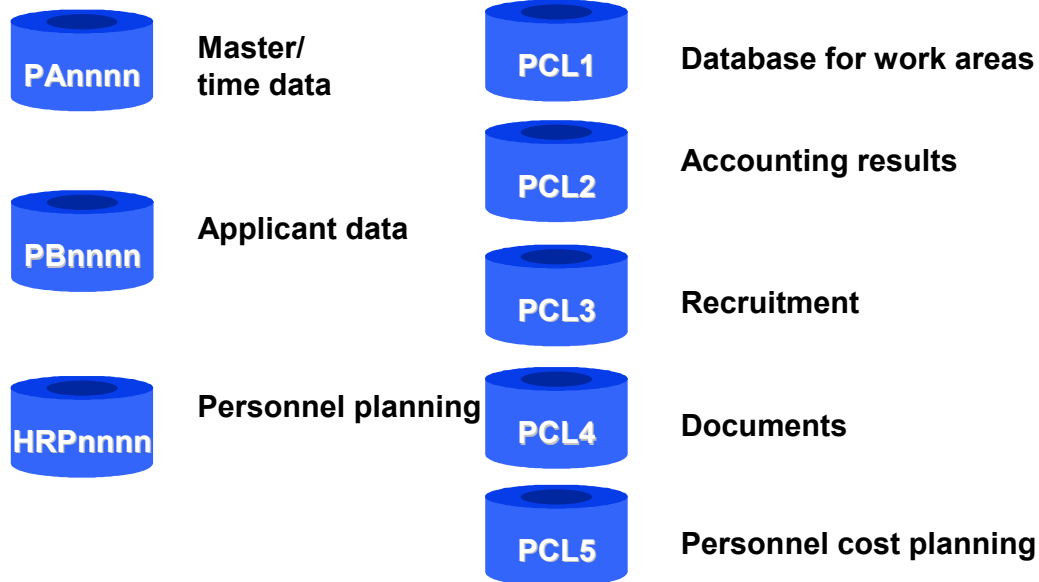


- You want to draw up your own evaluations of data in Personnel Administration. To do this, you must learn about data structures in Personnel Administration.



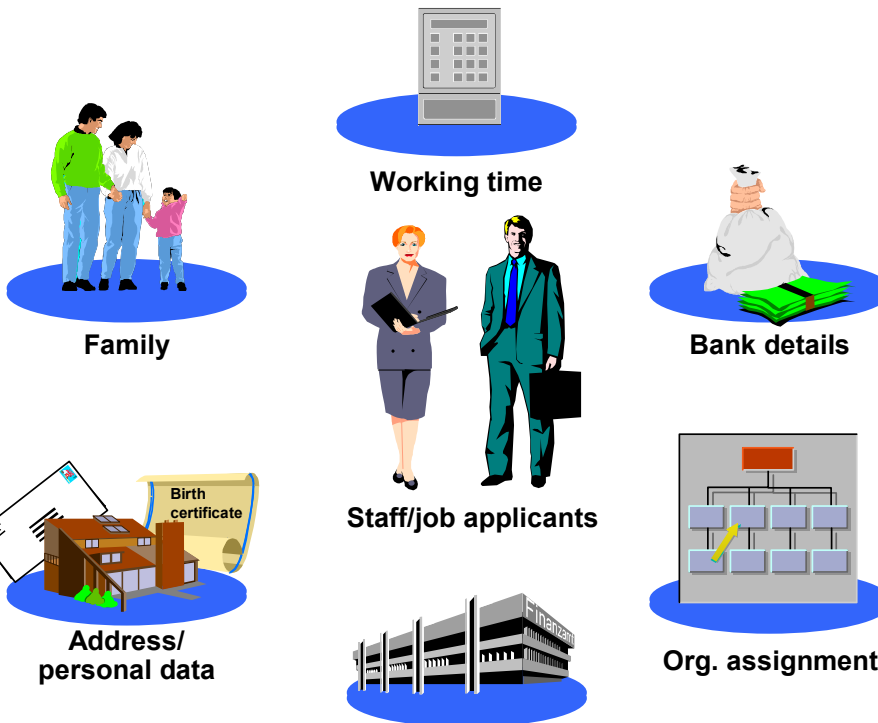
**At the conclusion of this topic, you will be able to:**

- **Describe the concept of infotypes in HR**



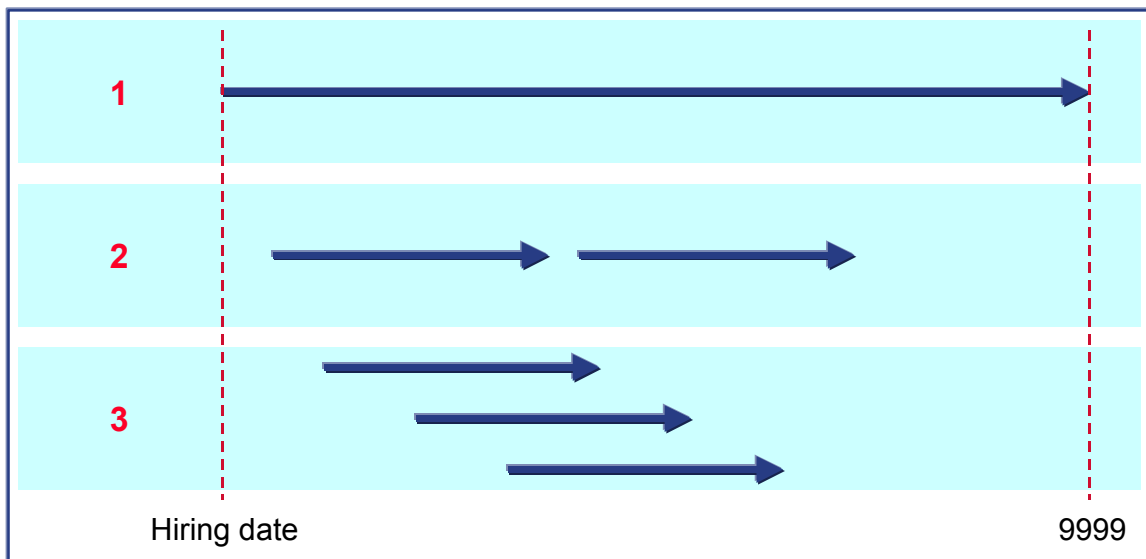
© SAP AG 2003

- Data structures in HR are a combination of master data, transaction data, and results data.
- The master data of Personnel Administration, Time Management, and Personnel Planning are structured in the same way.
- There are also different data clusters, which contain data from specific processing steps, such as time events on cluster PCL1 or accounting results on cluster PCL2.
- The infotypes are grouped into different **number ranges nnnn**:
  - 0000 – 0999 HR master data
  - 1000 – 1999 Personnel Planning
  - 2000 – 2999 Time Management
  - 4000 – 4999 Recruitment
  - 9000 – 9999 Customer-specific enhancements



© SAP AG 2003

- **Infotypes** are information units in the Human Resources Management system. Groups of related data fields are bundled into infotypes. Infotypes structure information, facilitate data entry and allow data to be saved for specific periods. Infotypes serve as templates for users to enter data. From a database point of view, infotypes provide a data structure and a set of coherent data records.
- The infotypes *Actions* 0000, *Organizational Assignment* 0001, *Personal Data* 0002, and *Payroll Status* 0003 are prerequisites for entering a personnel or applicant number.
- Save the infotypes at intervals to build up an infotype history. The system stores a validity period for each infotype record. Therefore, several data records usually exist for each infotype of an employee, and each record has a different validity period.
- You use a **time constraint** to determine how the data records of an infotype react to each other in terms of time.
- Several infotypes have **subtypes**, which break down information further. For example, technically speaking, the address types for infotype *Address* 0006 represent the subtypes.



© SAP AG 2003

- Each infotype and subtype has a time constraint.

- **Time constraint 1:**

Complete saving with no overlaps at any time, that is, one valid data record of the relevant infotype must exist during the duration of the employee's employment at the company.

- **Time constraint 2:**

Incomplete saving without overlaps at any time, that is, of an infotype may exist at a given time. No more than one valid data record of the infotype can exist at any given time.

- **Time constraint 3:**

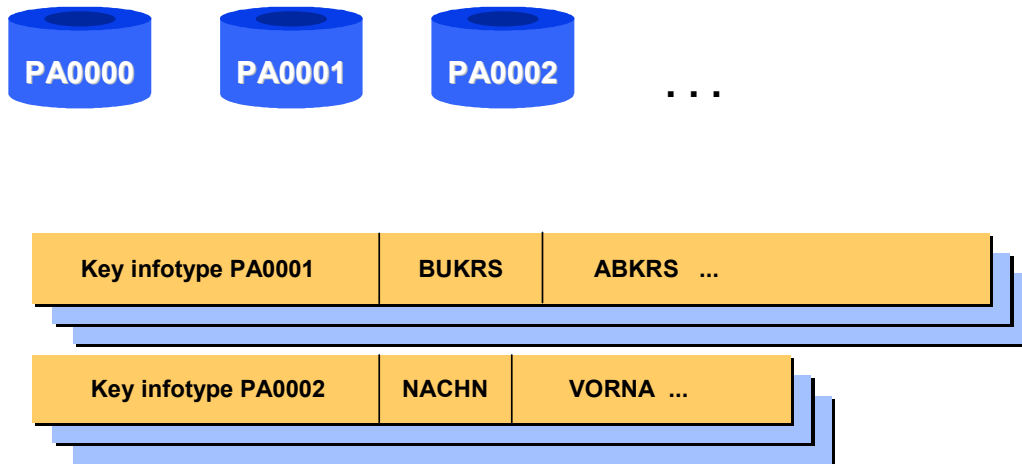
Incompleteness and overlaps permitted, that is, any number of valid data records of an infotype may exist at any given time.



**At the conclusion of this topic, you will be able to:**

- **Explain the technical data structures of the infotypes in Personnel Administration**

The structure of transparent tables:



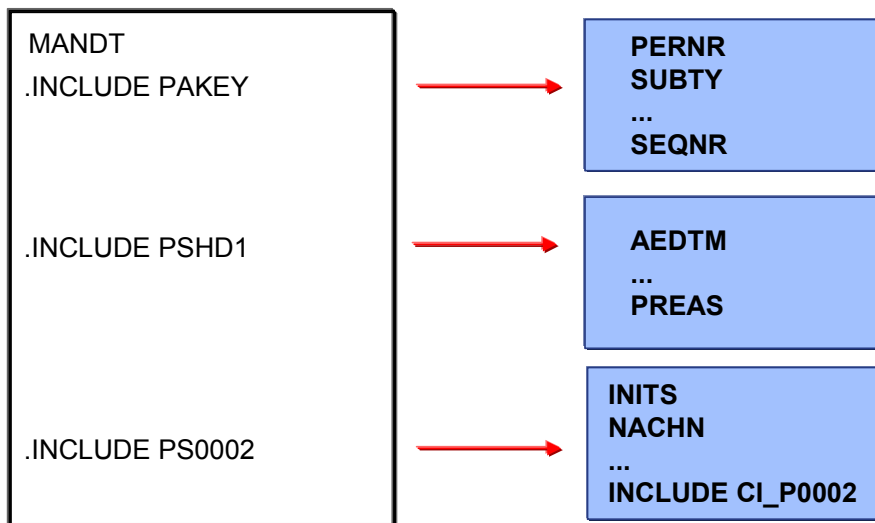
© SAP AG 2003

- The data records of the infotype **nnnn** are saved in the transparent database tables **PA#####**.
- The table's **primary key** refers to a minimum quantity of fields in a table, whose value uniquely identifies each data record of the table. In this context, minimum means that after any field is removed from the key, the remaining fields no longer uniquely identify the data records.
- The fields involved in a primary key are referred to as the table's **key fields**. This means that a value combination of the key fields may only occur once in a table.
- A **transparent table** is made up of the primary key and other non-primary keys. As regards infotype tables, non-primary fields include above all the infotype-specific fields, which contain the actual data or information on the infotype.



- Transparent tables *dictionary field definition*

Example: PA0002



© SAP AG 2003

■ Each PAnnnn infotype table and the client (MANDT field) is defined in the ABAP Dictionary using the following three include structures:

- **PAKEY:**  
Contains the infotype table key fields
- **PSHD1:**  
Contains administrative information on the infotype record (amongst other things, the date the last change was made and the name of the user who made the change)
- **PSnnnn:**  
Contains the infotype-specific fields that hold the infotype's actual information and data

**.INCLUDE PAKEY****PERNR****SUBTY****OBJPS****SPRPS****ENDDA****BEGDA****SEQNR**

© SAP AG 2003

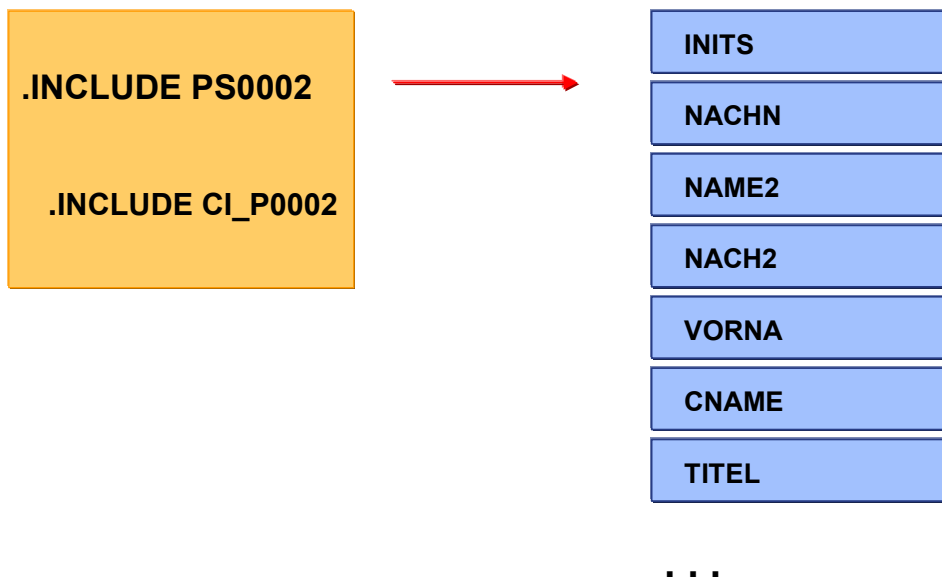
- The **PAKEY** structure contains an infotype's primary key. It is the same for all Personnel Administration infotypes and is made up of the following key fields:
- **PERNR**: In a client, the personnel number is the only unique key that identifies an employee. It can be used to display and maintain master data and working time data (infotypes) for an employee.
  - **SUBTY**: Subtypes are subdivisions of infotypes. The subtypes of an infotype may have different time constraints and create their own history.
  - **OBJPS**: The object identification indicator is used to differentiate records with the same infotype, subtype, lock indicator, valid from date, and valid to date. For example: Child number in infotype 0021 *Family Member/Dependents*.
  - **SPRPS**: The lock indicator for HR master data can be used to lock and unlock data records. It can be used to guarantee double control, that is, at least two users are involved in writing an active data record to the database. One of the users creates a locked infotype record. Another user unlocks this record by activating it.
  - **ENDDA**: Valid to date
  - **BEGDA**: Valid from date
  - **SEQNR**: The sequential number differentiates infotype records with the same key and the time constraint '3'. Unlike the personnel object identification, it is automatically assigned by the system.

**.INCLUDE PSHD1****AEDTM****UNAME****HISTO****ITXEX****REFEX****ORDEX****ITBLD****PREAS****FLAG1****FLAG2**

© SAP AG 2003

- The structure **PSHD1** is another basic structure that is the same for all Personnel Administration infotypes. Other basic administrative information for each infotype is saved here.
- This information includes the date the record was changed (**AEDTM**), the name of the person making the change (**UNAME**), and information on whether the administrator created plain text for this infotype (**ITXEX**).

Example: PA0002



© SAP AG 2003

- The structure **PSnnnn** contains the infotype-specific data fields, that is, the data fields that contain the actual infotype-specific characteristics.
- Each Personnel Administration infotype contains an include **CI\_Pnnnn** in the data structure PSnnnn. Customer-specific fields can be added to this include as enhancements to an infotype. In the standard SAP system, the include CI\_Pnnnn is empty.

## DB structure PAnnnn using infotype 0002 as an example



MANDT	PAKEY	PSHD1	PS0002
-------	-------	-------	--------

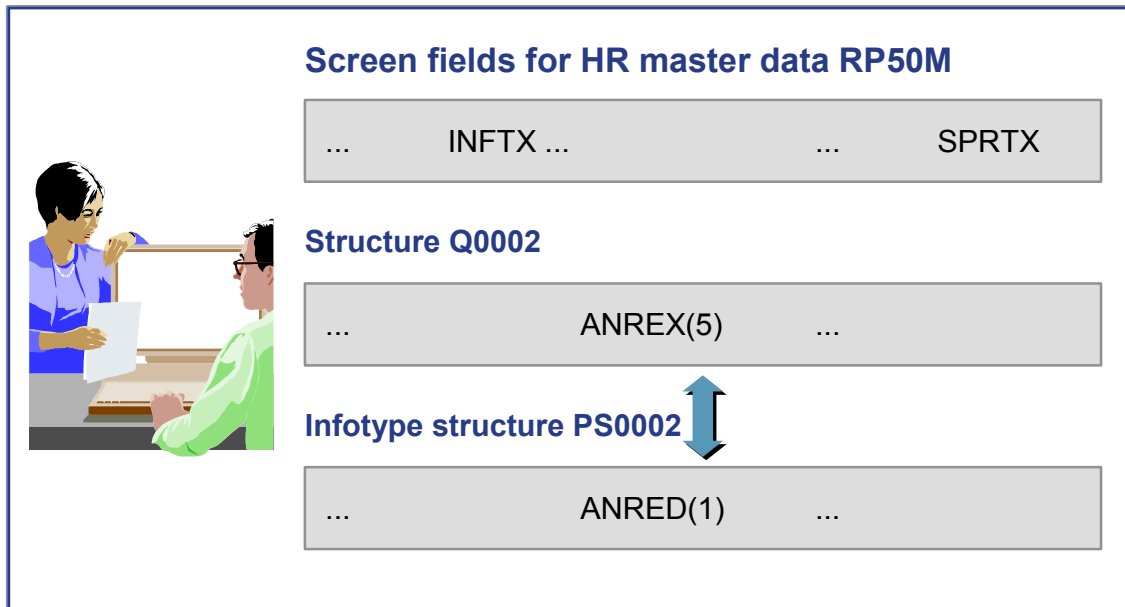
## Infotype structure Pnnnn using infotype 0002 as an example



PSKEY	PSHD1	PS0002
-------	-------	--------

© SAP AG 2003

- During interface definition, the logical structure **Pnnnn** is used between programs and between programs and screens.
- The key **PSKEY** contains the key fields of the structure **PAKEY**, and an additional field **INFTY** for the infotype number **nnnn**. The client field is not required in this structure.
- The structure **PSnnnn** is also included by the structure **Pnnnn** so that the transparent table **PAnnnn** and the logical structure **Pnnnn** are always consistent with each other.



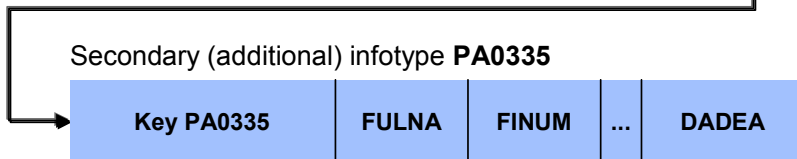
© SAP AG 2003

- Fields (such as header lines) that are displayed for all infotypes are stored in the structure RP50M.
- The structure **Qnnnn** also exists for several infotypes. It contains screen fields for data entry. An employee's form of address, for example, is stored with a numeric key in the field PA0002-ANRED. The screen, however, must enable the form of address itself to be displayed and entered, not its numeric key. For this reason, the additional field Q0002-ANREX is used.

- **Infotype view consisting of a primary and secondary (additional) infotype**

Primary infotype **PA0021**

Key PA0021	FAMSA	FGBDT	...	EMRGN
------------	-------	-------	-----	-------



**Tables T582V/T582W:** View definition - assignment of additional infotype to primary infotype

© SAP AG 2003

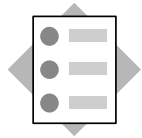
- Many country versions require country-specific fields that are attached to all international infotypes, such as *Organizational Assignment*, *Family Member/Dependents*.
- To prevent international infotype structures being crammed with country-specific fields, these fields have been moved to **additional infotypes**.
- Whereas international infotypes ("primary infotype") have the value *I*, additional infotypes have the value *Z* in the *Infotype/View* field in the table T777D.
- The primary infotype and the additional infotype are maintained together on one screen in an **infotype view** (defined using the tables T582V and T582W). Then the infotypes are saved, the data you are maintaining is distributed across two infotype records (one primary and one additional infotype record) with an identical infotype key.
- The infotype menu only contains the international primary infotype. The additional infotype can only be maintained with the primary infotype, and not on its own. All infotype characteristics such as the time constraint are only defined on the primary infotype.
- When a personnel number is being created, the attribute IVWID is used to assign a fixed view indicator in the infotype *Payroll Status* 0003. The secondary infotypes are selected using this view indicator. This view indicator must NEVER be changed, for obvious reasons.

Element	Definition	Included	Changeability
<b>PAKEY</b>	<b>Structure</b>	-	<b>No</b>
<b>PSHD1</b>	<b>Structure</b>	-	<b>No</b>
<b>PSnnnn</b>	<b>Structure</b>	<b>CI_Pnnnn</b>	<b>PS0nnn-PS8nnn: Only CI_Pnnnn PS9nnn: Yes</b>
<b>Pnnnn</b>	<b>Structure</b>	<b>PSKEY PSHD1 PSnnnn</b>	<b>P0nnn-P8nnn: No P9nnn: Only PS9nnn</b>
<b>PAnnnn</b>	<b>Transparent table</b>	<b>MANDT PAKEY PSHD1 PSnnnn</b>	<b>PA0nnn-PA8nnn: No PA9nnn: Only PS9nnn</b>
<b>CI_Pnnnn</b>	<b>Structure</b>	-	<b>Yes</b>

© SAP AG 2003

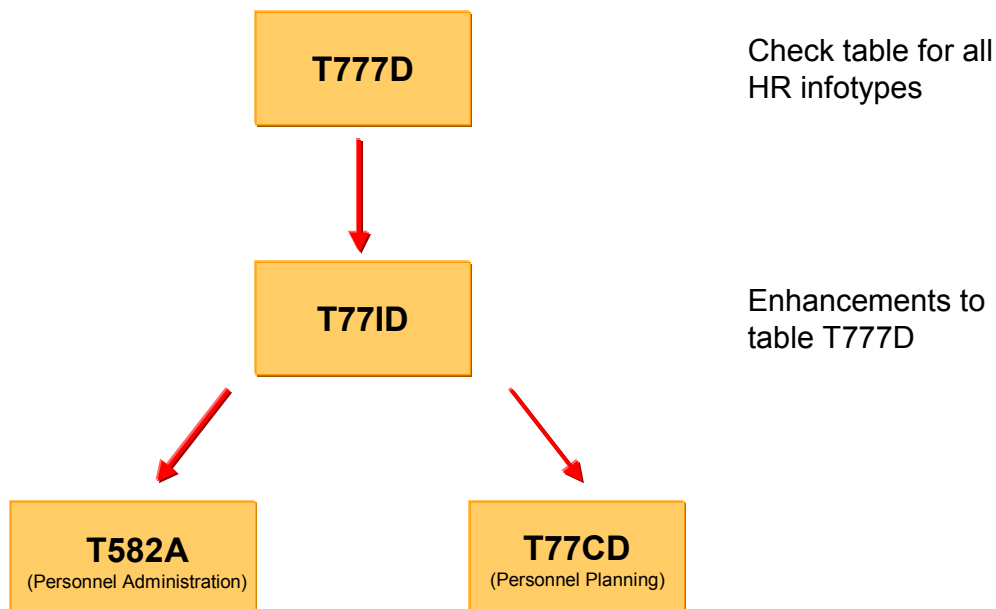
- The table displayed above provides an overview of the data structures and transparent tables that belong to an infotype in Personnel Administration.





**At the conclusion of this topic, you will be able to:**

- **Use the technical information contained in the check and control tables of HR infotypes**



© SAP AG 2003

- The table *Infotypes - Dialog/Database Assignment* (T777D) is the key check table (domain INFOTYP) for all HR infotypes (administration and planning infotypes). This table contains the names of all infotype-dependent repository objects (such as tables, structures, and programs).
- The table *Infotypes - Enhancements to T777D* (T77ID) supplements the table T777D that exists for reasons of memory space.
- The table *Infotypes - Customer-Specific Settings* (T582A) is used to make customer-specific settings for administration infotypes.
- The table *Infotypes - Customer-Specific Settings* (T77CD) can be used to make customer-specific settings for planning infotypes.
- You can use the report RHT777DCHECK to check the consistency of entries in the specified tables.

- Infotype control

Table T777D

Infotype	Structure	DB Table	DB Table Table Infotype
0000	P0000	PA0000	
0001	P0001	PA0001	
...			
1000	P1000	HRP1000	
1001	P1001	HRP1001	
1002	P1002	HRP1002	HRT1002
...			
9998	P9998	PA9998	
9999	P9999	HRP9999	

© SAP AG 2003

- The table *Infotypes - Dialog/Database Assignment* (T777D) is automatically maintained using the tool for creating new infotypes. Do not edit this table manually.
- In addition to the transparent tables **PA#####** and **HRP#####**, each infotype in the Human Resources Management system has a **logical structure P#####** (stored in T777D-PPNNN).
- Logical structures are used to define interfaces between programs and between screens and programs. No data records exist on the database for a logical structure.
- To avoid direct database accesses, these logical structures P##### are primarily used in Human Resources Management programs.

**Table T777D**

Infotype	Structure	Infotype-Spec. Fields	Prim. Table Ext. Infotype
0000		PS0000	
0001		PS0001	
...			
1000		HRI1000	
1003		HRI1003	
...			
1611		HRI1611	V_T5U27
1612		HRI1612	V_T5U26
...			
9998		PS9998	
9999		HRI9999	

© SAP AG 2003

- The table *Infotypes - Enhancements to T777D* (T777ID) only supplements the *Infotypes - Dialog/Database Assignment* table T777D, for reasons of memory space.
- This table includes the structure PSnnnn or HRInnnn for infotype-specific fields for each infotype.

- Control applicant infotypes

**Table T777D**

Infotype	DB Table	DB Table Applicant Infotype
0000	PA0000	PB0001
0001	PA0001	PB0002
0002	PA0002	
...		PB0006
0006	PA0006	PB0007
0007	PA0007	
...		PB4000
4000		PB4001
4001		PB4002
4002		

© SAP AG 2003

- Infotype records for applicant infotypes are stored in the transparent tables **PBnnnn**, which are structured in the same way as PAnnnn tables.
- The following types of applicant infotypes exist:
  - Infotypes that can be maintained for employees and applicants (for example, 0001, 0002, 0006, and 0007). Here, there is a database table **PAnnnn** for employee data and a database table **PBnnnn** for applicant data.
  - Infotypes that may only be maintained for applicants (applicant infotypes only, with infotype numbers from 4000 to 4999). Here, there is only one database table solely for applicant data (**PB4nnn**).

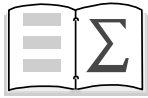
- Subtype control

Table T777D

Infotype	Subtype Field	Subtype Table	Subtype Text Tab.	TC Table
0006	ANSSA	T591A	T591S	T591A
...				
0008	SUBTY	T591A	T591S	T591A
0009	BNKSA	T591A	T591S	T591A
0010	LFDNR	T591A	T591S	T591A
0011	LGART	T512Z	T512T	T591B
...				
0019	TMART	T531	T531S	
...				
0022	SLART	T517T	T517T	

© SAP AG 2003

- *Subtypes* are used to further break down *infotypes*, for example, to further classify infotype 0006 *Addresses* into main address, second address, business address, and so on.
- In the table T777D, the following fields are used to define subtypes:
  - **Subtype field:** Here, the field is in the infotype structure that contains the subtype. Usually this is the field *SUBTY*. However, it may also be a field other than *SUBTY* and contain the value of the subtype (for example, *ANSSA* = address type in the infotype 0006).
  - **Subtype table:** This contains the check table that contains all allowed subtype values for each infotype. This is usually the table *T591A*. However, the subtype values may also be stored in another table - depending on the subtype field - for example, the table *T512Z* for infotype 0011 *Ext. Bank Transfers* (permissible wage types for each infotype), as the wage type (field *LGART*) is also the subtype.
  - **Subtype text table:** This lists the text table for the subtype table, that is, the table with the names for the individual subtype values. This is usually the table *T591S*, if the table *T591A* is used as a subtype table. If another subtype table is used (for example, *T512Z*), the corresponding text table (for example, *T512T*) is stored here.
  - **Time constraint table:** If the time constraint is defined based on the subtype, the corresponding time constraint table appears here (for example, *T591A/T591B*).



**You are now able to:**

- **Describe the data structures of Personnel Administration**
- **Use the check and control tables of HR infotypes as a source of information**





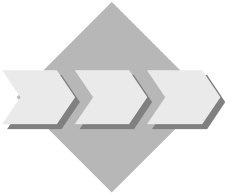


## Unit: Data Structures in Personnel Administration



At the end of this exercise, you will be able to:

- Describe the data structures of Personnel Administration
- Use the check and control tables of HR infotypes as a source of information



You want to draw up your own evaluations of Personnel Administration data. To do this, you must learn more about data structures in Personnel Administration.

- 1-1 The fields SPRPS and OBJPS are located in the key of Personnel Administration infotypes. Write down the functions of both fields.

1-1-1 **SPRPS:**

---

---

1-1-2 **OBJPS:**

---

---

- 1-2 What tasks do the structures Qnnnn and CI\_Pnnnn fulfil?

1-2-1 **Qnnnn:**

---

---

1-2-2 **CI\_Pnnnn:**

---

---

- 1-3 Several infotypes have subtypes. If this is the case, the infotype's time constraint is on the subtype.

1-3-1 How do you determine which time constraint is assigned to the subtype of an infotype?





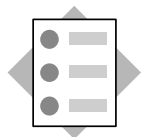
## Unit: Data Structures in Personnel Administration

- 1-1 The fields SPRPS and OBJPS are located in the key of Personnel Administration infotypes. Write down the functions of both fields.
- 1-1-1 **SPRPS:**  
If this field is not blank, the infotype record is locked. It needs to be unlocked to be activated.
- 1-1-2 **OBJPS:**  
This field is used to differentiate infotype records with the same infotype, subtype, block indicator, valid from date, and valid to date.
- 1-2 What tasks do the structures Qnnnn and CI\_Pnnnn fulfil?
- 1-2-1 **Qnnnn:**  
The structure Qnnnn contains screen fields with different lengths than the corresponding database fields.
- 1-2-2 **CI\_Pnnnn:**  
The customer-specific fields of a standard infotype are grouped in the structure CI\_Pnnnn.
- 1-3 Several infotypes have subtypes. If this is the case, the infotype's time constraint is on the subtype.
- 1-3-1 To establish the time constraint of the infotype's subtype, determine which time constraint table is assigned to the infotype in the table T777D. View the table.



## **Contents:**

- **Functions of the logical database**
- **Data retrieval**
- **Processing of infotypes**
- **Period-based data**
- **Person selection and data selection**
- **Sort order**
- **Setting up the selection screen**
- **Selection view for free delimitations**
- **The logical database PNPCE**



**At the conclusion of this unit, you will be able to:**

- **Explain the functions and data structures of the logical database PNP**
- **Process period-based data in a report**
- **Use the screening criteria to select data and persons**
- **Set up the standard selection screen**
- **Create a selection view for free delimitations**
- **Use the logical database PNPCE in PNP mode**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration

Data Structures in Personnel Planning

 The Logical Databases PNP/PNPCE

The Logical Database PCH

Join and Projection

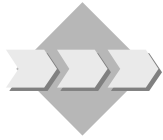
Customer-Specific Infotypes

Specific Commands

HR ABAP Features

Repetitive Structures and List Display

© SAP AG 2003



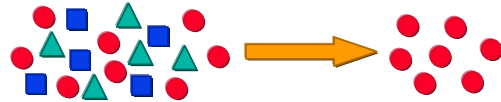
- **The Personnel Department in your company requires its own reports to evaluate HR master data.**
- **As a member of the IT team, it is your responsibility to program these reports.**
- **You would like to make optimal use of the functions offered by the logical databases in HR.**



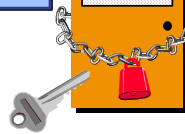
## 1. Data retrieval



## 2. Screening



## 3. Authorization check



© SAP AG 2003

- To minimize the amount of programming required, it often makes sense to use logical databases when generating reports. Reports are special programs that provide selection screens and access databases. The logical database PNP consists of the database driver **SAPDBPNP** and is activated using the report attributes. It fulfils three main functions:
  - **Data retrieval:**  
HR data is written for each employee to the main memory, where it can be processed.
  - **Screening:**  
A selection screen enables you to select employees in accordance with organizational criteria. For example, you can select all of the hourly wage earners of a particular personnel subarea.
  - **Authorization check:**  
The implicit authorization check is another advantage of the logical database. Personnel data is frequently confidential and it does not make sense to allow all programs carry out the authorization check.

**1. Create data structures for infotypes**

```
TABLES: PERNR.  
INFOTYPES: 0001,  
            0002,  
            0007.
```



"Organizational Assignment"  
"Personal Data"  
"Planned Working Time"

**2. Fill the data structures with the infotype records**

```
GET PERNR.
```

© SAP AG 2003

- The structure **PERNR** must be declared with the **TABLES** statement.
- The **INFOTYPES nnnn** statement is used to create an internal table with the name Pnnnn for each declared infotype.
- When the **GET** event occurs, the tables of declared infotypes are filled with all of the records that exist for a personnel number. The field **PERNR** in the structure **PERNR** contains the personnel number. For purposes of processing, you can access the **PERNR-PERNR** field. This field contains the formatted name of the employee.
- **Note:**  
When the **GET** event occurs, all internal infotype tables are filled. Therefore, only declare the infotypes you wish to process. You can, however, use the enhancement **MODE N** for the statement **INFOTYPES** to suppress the filling of internal tables.

```

TABLES: PERNR.
INFOTYPES: 0000,      "Actions
            0002,      "Personal Data
            0006,      "Addresses
            ....

GET PERNR.
  LOOP AT P0002 WHERE ENDDA GE PN-BEGDA
              AND  BEGDA LE PN-ENDDA.
    WRITE...
  ENDLOOP.
    
```

© SAP AG 2003

- When the GET PERNR event occurs, the infotype records are imported to internal tables Pnnnn (for example, P0006 for infotype 0006). These tables may then be processed in a **LOOP-ENDLOOP** loop.
- All infotype records whose validity period overlaps the period selected in the selection screen (PN-BEGDA and PN-ENDDA) by at least one day are placed one after the other in the header of the respective infotype table Pnnnn.
- For example, if you choose the current year in the selection screen, PN-BEGDA contains the first day of the year and PN-ENDDA contains the last day of the year. If you do not enter any data for the period in the selection screen, PN-BEGDA contains the low date (January 01, 1800) and PN-ENDDA contains the high date (December 31, 9999).
- **Note:**  
Infotypes with the **time constraint 3** must be processed in this way.

```
TABLES: PERNR.  
INFOTYPES: 0001,      "Actions  
            0002,      "Personal Data  
            0006,      "Addresses  
            ....  
  
GET PERNR.  
  PROVIDE * FROM P0002  
    BETWEEN PN-BEGDA AND PN-ENDDA.  
  WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- Infotypes with the time constraint 1 or 2 may be processed with a **PROVIDE-ENDPROVIDE** loop.
- All infotype records whose validity period overlaps the period selected in the selection screen (PN-BEGDA and PN-ENDDA) by at least one day (BETWEEN-AND) are placed one after the other in the header of the respective infotype table Pnnnn.
- **Note:**  
If you have entered period dates in the selection screen, the contents of the fields Pnnnn-BEGDA and Pnnnn-ENDDA are also reset in the **header** of the infotype table Pnnnn. For example, if the date in Pnnnn-ENDDA is after the date in PN-ENDDA, Pnnnn-ENDDA is given the value from PN-ENDDA.

```
TABLES: PERNR.  
INFOTYPES: 0000,      "Actions  
            0002,      "Personal Data  
            0006,      "Addresses  
            ....  
  
GET PERNR.  
  PROVIDE * FROM P0006  
    BETWEEN PN-BEGDA AND PN-ENDDA  
    WHERE P0006-SUBTY = '1'.  
  WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- If the infotype to be maintained has subtypes, it must be processed with the subtype. This is brought about by the **WHERE** condition in the PROVIDE statement. In the above example, processing of the infotype *Addresses* (0006) is restricted to subtype 1 *Permanent Address*. This prevents the intervals from overlapping. This is required as the PROVIDE processing only occurs correctly for infotype records with periods that do not overlap.

```
TABLES: PERNR.  
INFOTYPES: 0002,      "Personal Data  
            0015,      "Additional Payments  
            ....  
  
GET PERNR.  
CHECK LINES( P0015 ) GT 0.  
PROVIDE * FROM P0002  
    BETWEEN PN-BEGDA AND PN-ENDDA.  
    WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- Frequently, you only want to display employees with data of a specific infotype.
- You can use the **CHECK LINES( *Pnnnn* )** statement to determine the number of filled lines of an infotype table. The subsequent statements are only executed if the condition is fulfilled.

```

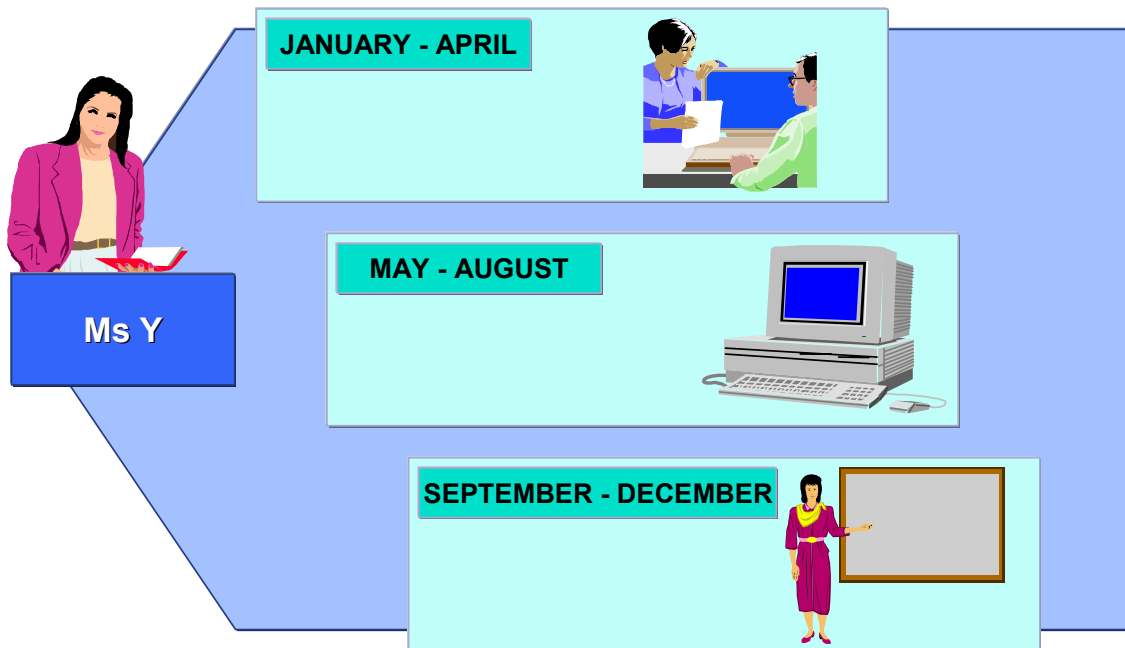
GET PERNR.
  LOOP AT P0002 WHERE ENDDA GE PN-BEGDA
           AND  BEGDA LE PN-ENDDA.
    WRITE...
  ENDLOOP.
or:
  PROVIDE * FROM P0002
    BETWEEN PN-BEGDA AND PN-ENDDA.
    WRITE...
  ENDPROVIDE.

  PROVIDE * FROM P0006
    BETWEEN PN-BEGDA AND PN-ENDDA
    WHERE P0006-SUBTY = '1'.
    WRITE...
  ENDPROVIDE.

END-OF-SELECTION.
    
```

© SAP AG 2003

- HR data is processed in two nested loops:
- A GET PERNR loop for all of the personnel numbers selected. It is concluded implicitly by the next event, for example, END-OF-SELECTION.
- Subordinate loops per infotype, for the processing of all infotype records for the selected personnel number.
- With this form of processing, you should note that the data for the infotypes *Personal Data* (0002) and *Addresses* (0006) is listed sequentially and is not linked.



© SAP AG 2003

- Infotype data is period-based, in other words, it is only valid for specific periods. For this reason, each record has a valid from date and a valid to date.
- This example shows the jobs that an employee has performed over the course of a year.
- The decision on how to retrieve data is made for each individual infotype.



**Data  
selection**



**Org.  
assignment**



**Org.  
assignment**



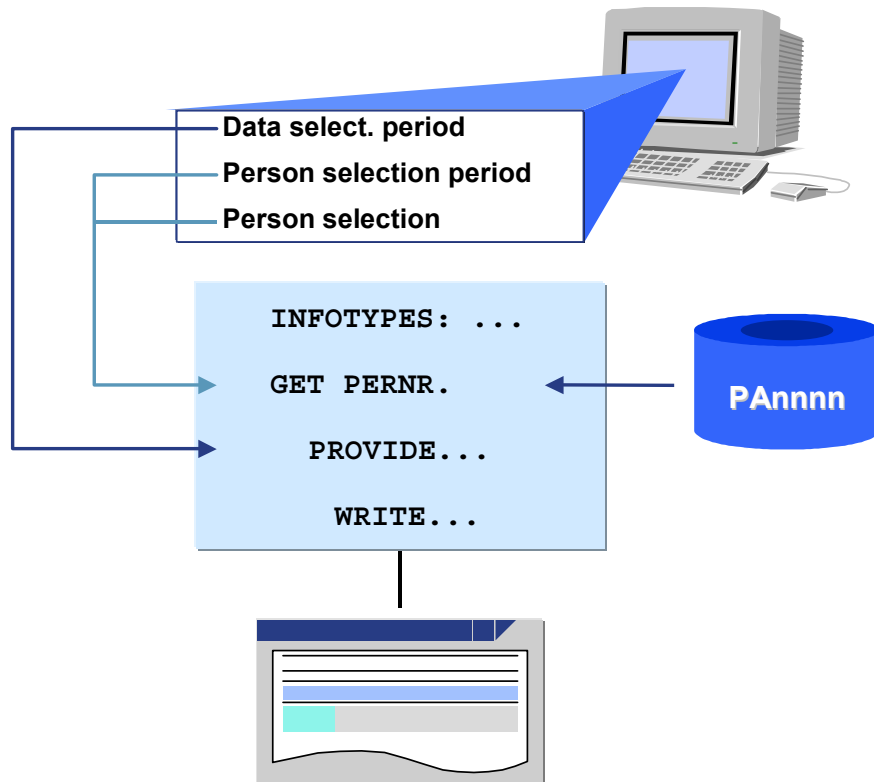
© SAP AG 2003

- Both types of data selection are based on the data selection period entered in the selection screen.
- Processing with LOOP-ENDLOOP and PROVIDE-ENDPROVIDE provides data for a specific period, that is, according to the data selection period.
- Frequently, only the most recent or oldest infotype record is required from the selected period, rather than all of the infotype records. The RP\_PROVIDE\_FROM\_LAST and RP\_PROVIDE\_FROM\_FRST macros can be used in such situations.



**At the conclusion of this topic, you will be able to:**

- **Use the screening criteria**
- **Define default values for the selection screen in your report**
- **Sort personnel numbers according to organizational criteria**
- **Set up the standard selection screen to suit a user's needs**
- **Create a selection view for free delimitations**



© SAP AG 2003

- When data is screened, a distinction is made between two types of selection: Person selection, and the data and person selection period.
- The system uses the person selection and the person selection period to determine which employees correspond to the selection criteria, for example, the monthly wage earners for a specific personnel subarea.
- The data selection period specifies the period for which the data contained in the infotype tables is evaluated by the report.
- The selection screen enables you to change the sort sequence for HR records.

Period

☒ Today
 ☐ Current month
 ☐ Current year

☐ To current date
 ☐ From today

☐ Other period

Data select. period  to

Person selection period  to

Payroll period

GET PERNR.

PROVIDE \* FROM Pnnnn  
 BETWEEN PN-BEGDA  
 AND PN-ENDDA.  
 ...  
 ENDPROVIDE.

© SAP AG 2003

- The PN structure (defined in DBPNPCOM) that is available in the report is populated from the selection screen. The data from the start and end dates of the **data selection period** is transferred to the fields PN-BEGDA and PN-ENDDA. If you use a data selection period, infotype data records are entered in the PROVIDE loop if their validity periods overlap with the period selected by at least one day.
- The start and end dates of the **person selection period** that are entered by the user are available in the fields PN-BEGPS and PN-ENDPS. This interval is the criterion used for selecting the personnel numbers to be processed. All personnel numbers that fulfil the selection criteria specified (see overleaf) at any given time in the person selection period entered by the user, are used in processing.
- If the period is specified using radio buttons, the fields PN-BEGDA/ENDDA and PN-BEGPS/ENDPS contain the same period. In the example displayed above, *Today* is set as the key date. In this case, the four fields contain the system date.
- Note that all of the data records for an infotype are available in the corresponding internal table for GET PERNR. In other words, data selection does not restrict the ability to read from the database.

## Selection

Personnel number



Employment status



Company code



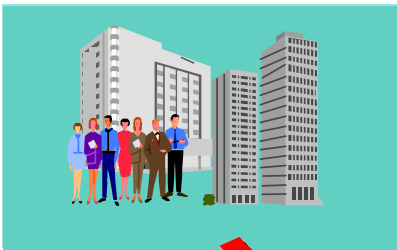
Payroll area



Pers.area/subarea/cost center



Employee group/subgroup



GET PERNR.



© SAP AG 2003

- The objective and result of person selection is a group of employees for whom an evaluation is created.
- You can select persons by entering the following criteria, for example:- A range of personnel numbers- Characteristics of the organizational assignment- The status of the employee, for example, whether s/he is active or inactive
- All the criteria that the user enters here must be fulfilled by the personnel numbers to be selected in the specified person selection period. The data from the logical database is only made available to your report when the event GET PERNR occurs if these criteria are fulfilled.

Period

☐ Today
☐ Current month
☒ Current year

☐ To current date
☐ From today

☐ Other period

to

Selection

Personnel number

Payroll area

## INITIALIZATION.

```

PNPTIMED          = 'Y' .
PNPABKRS-LOW      = 'D1' .
PNPABKRS-HIGH     = 'D2' .
PNPABKRS-OPTION   = 'BT' .
PNPABKRS-SIGN     = 'E' .
APPEND PNPABKRS.
    
```

© SAP AG 2003

- When the INITIALIZATION event occurs, you can set default values for the selection screen.
- Default radio buttons for selecting the period can be set by transferring one of the following values to the field PNPTIMED:
  - D = Today (key date)
  - M = Current month
  - Y = Current year
  - P = Up to today (from past to current date)
  - F = As of today (from the current date into the future)
- The person selection fields are defined in the include program **DBPNPSEL**. They are internal tables that must be filled using APPEND.

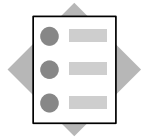


...  
GET PERNR.  
...

PerNo.	Pers.Area	Name
1909	1000	Sam Hawkins
1910	1000	David Lindsay
1899	1100	Karl May

© SAP AG 2003

- The standard sort sequence is in ascending order by personnel number.
- You can use the sort function to sort an evaluation using organizational criteria.
- For example, you can use the sort function to create a hierarchical list according to personnel subarea and administrator, or to list employees in alphabetical order.
- The sort function can be used for all of the fields in infotype 0001 *Organizational Assignment*. You can also determine the sort sequence.
- Different sort types are available for evaluations of specific periods.
- If data is selected by matchcode, personnel numbers are sorted by matchcode sequence.



**At the conclusion of this topic, you will be able to:**

- **Use the logical database PNPCE in PNP mode**



- **Employee as person with multiple employment with several employment contracts**
- **Separate personnel number for each contract**
- **Assignment of individual personnel numbers to a central person (object type CP)**
- **Additional external person ID in the infotype 0709 *Person ID***

© SAP AG 2003

- As of the SAP R/3 Enterprise Release, **Concurrent Employment** allows for employment of persons with multiple employment and several concurrent employment contracts with a company.
- A separate personnel number is then assigned for each contract. The multiple employments are grouped by assigning the corresponding personnel numbers to a **central person** (object type CP).
- An external **Person ID** is also determined and stored in infotype 0709 *Person ID*.
- In order to enable the concurrent employment function, you must use the logical database PNPCE. However, when you use new developments in the SAP R/3 Enterprise Release and do not use the concurrent employment function, we recommend that you use this new logical database and improved selection screen as it allows you to use more detailed settings in the report classes.

```
TABLES: PERNR.
NODES: PERAS.
INFOTYPES: 0000,      "Actions
            0002,      "Personal Data
            0006,      "Addresses
            ....

GET PERAS.
  PROVIDE * FROM P0002
    BETWEEN PN-BEGDA AND PN-ENDDA.
  WRITE...

ENDPROVIDE.
```

© SAP AG 2003

- In the report, the structure PERNR must be declared by the TABLES statement. However, further use of the structure PERNR is restricted. For example, the event GET PERNR is not permitted. Use **GET PERAS** instead. Except for the component PERNR-PERNR, all other components of the structure PERNR are no longer populated and have initial values. Therefore, programming on these values (with the exception of PERNR-PERNR) is not permitted.
- The events GET GROUP and GET PERSON are also available. When these events occur, you must declare them using NODES. If these events do not occur, it is assumed that the concurrent employment function is not required. The report then runs in PNP mode.
- Only a few companies are currently using the concurrent employment function, and further details are therefore not provided here. For more information on using the concurrent employment mode, refer to the documentation for the program SAPDBPNPCE.



**At the conclusion of this topic, you will be able to:**

- **Set up the selection screen of the logical database to suit customer-specific needs**

Report category **00000350** **HR350**

☐ For the logical database PNPCE

☐ Data selection period = Person selection period

☐ Matchcode allowed    ☐ Sort allowed    ☐ Org.str. allowed

Data select. period  
☐ No entry  
☐ Key date  
☒ Interval

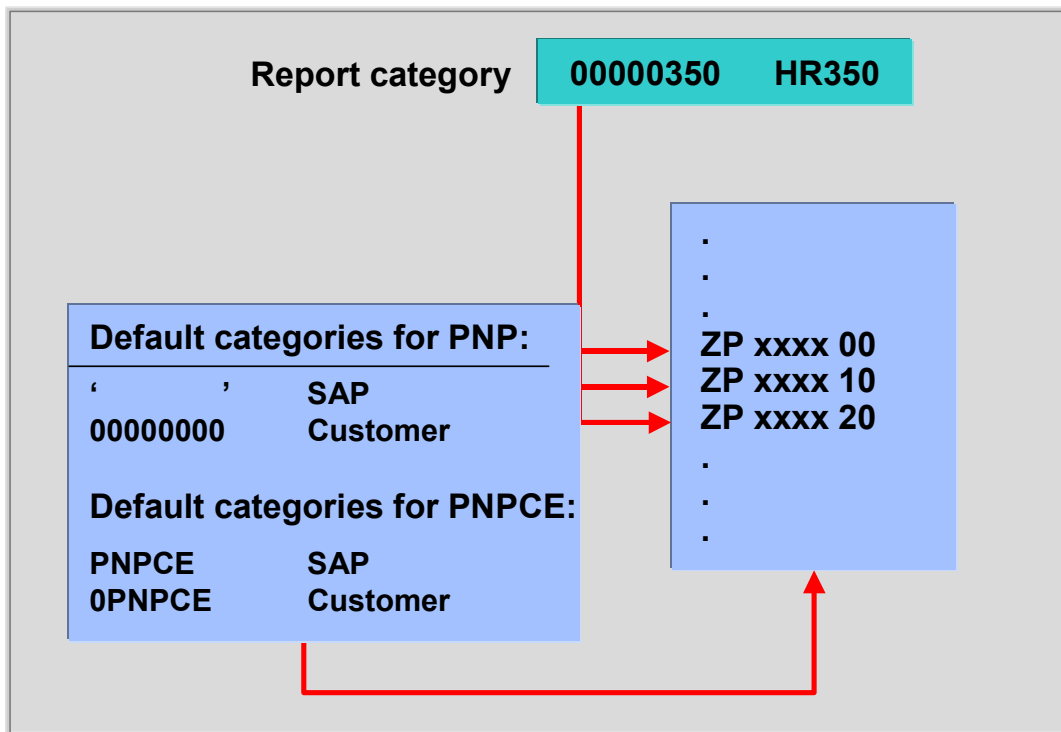
Person selection period  
☐ No entry  
☐ Key date  
☒ Interval

Payroll area/period/year  
☒ No entry  
☐ Entry possible  
☐ Mandatory field

Type **SAP**  
 Name **PNP\_DEFAULT**

© SAP AG 2003

- Think about which report categories you need. The customer-specific report categories must start with a number. Make a new entry.
- Select the field *Data selection period = Person selection period* if the person selection period is to adopt the value of the data selection period and is not to be ready for input.
- Determine whether sorting, selection using matchcodes, and access via the organizational structure are permitted.
- Decide whether the period determination indicator and the fields for the input of the data and person selection period should be displayed on the screen. Also specify whether it is possible to enter the payroll period.
- Choose a selection view for free delimitations. You can use either standard or customer-specific selection views. If no free delimitation can be selected on the selection screen, enter the view PNP\_NO\_FREE\_SEL.
- You can select the required selection fields after double-clicking *Permitted Selection Criteria* in the *Dialog Structure* group box. Select the fields that you want to appear on the first page of the selection screen. Fields that are not selected are displayed in an additional window when you choose *Further Selections*.



© SAP AG 2003

- In the standard SAP system, a report class is assigned to the reports. If you would like to override this class, carry out the steps described here. Making an assignment to report classes for standard reports using SE38 constitutes a system modification.
- For reports that you have developed yourself, you can make the assignment to report classes either using the Customizing activity described here (client-specific) or directly using the report attributes in SE38 (for all clients).
- If a report uses the logical database PNP, you can access the HR report category maintenance functions from the *Program Attributes* screen by choosing *Edit -> HR Report Category*. The report category currently assigned to the report is displayed in the dialog box. If no report category is assigned to this report in Customizing, the report category assigned by SAP is displayed.
- You can also use the SAP report categories for your customer reports. For example: You have created your own report ZPCTEST1, and you want the selection screen to correspond to the selection screen used in the evaluation reports for payroll that have the report category '\_\_M00001'. To do this, assign the report category '\_\_M00001' to your report.

1.

Create Selection View

Origin of view

CUS

☐ For logical database

☒ For any tables

Name of view

PNP\_VIEW\_1

2.

Tables

PA0002

PA0006

Other tables

© SAP AG 2003

- **Free delimitation** is one option you can use to add selection criteria to the selection screen. When you use this option, the additional selections are made directly by the logical database and performance is therefore significantly better than if you restrict the employees to be selected in the report itself.
- Call the **Object Navigator** (SE80) and from the *Workbench* menu, choose *Edit object*. Under *More...*, choose *Selection view* and choose *Create*.
- Enter a name for your view on the screen that appears. Select a name with the prefix **PNP**.
- On the next screen, enter the names of the database tables whose fields you want to use for the selection view.

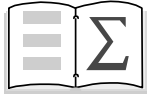
Functional groups		Tables/Nodes	
01	Personal data	PA0002	HR Master Record Infotype 0002
02	Address data	PA0006	HR Master Record Infotype 0006

Table/Node Fields		
02	STRAS	Street and house number
02	ORT01	City
02	ORT02	District
02	PSTLZ	Postal code
02	LAND1	Country code
02	TELNR	Telephone number

© SAP AG 2003

- Within a selection view, fields are selected and grouped together as **functional groups**. All fields assigned to a functional group are included in the view. By grouping the fields into functional groups, you can keep fields with similar contents together. This enables you to use the view to search for specific information.
- A functional group is defined by assigning a functional group identifier and a corresponding text. The function group identifier can consist of any two characters and is only relevant for maintaining the selection view.
- From the *Table/Node* window, select the table whose fields you want to assign to one of your defined functional groups. The system lists the fields in the *Fields in Table/Node* screen from where you can assign the required fields to a functional group.



**You are now able to:**

- **Explain the functions and data structures of the logical database PNP**
- **Process period-based data in a report**
- **Use the screening criteria to select data and persons**
- **Set up the standard selection screen**
- **Create a selection view for free delimitations**
- **Use the logical database PNPCE in PNP mode**





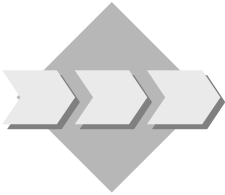
## Unit: The Logical Databases PNP/PNPCE

### Topic: Creating a List of Employees



At the end of this exercise, you will be able to:

- Process infotype records using **PROVIDE...ENDPROVIDE**



The personnel department in your company requires a list of employees specifying language and date of birth.

1-1 Create a report which generates a list of employees with the following information:

- Personnel number
- Name
- Language
- Date of birth

1-1-1 In the selection screen for the logical database, set the period determination indicator 'Today' as the default value.

1-1-2 Enable the selection of employee according to language using **SELECT-OPTIONS**.

1-1-3 Take the formatted names from the field **PERNR-ENAME**.

1-1-4 Separate the resulting field contents in your list by a vertical line (**SY-VLINE**).



Create your report using the name **ZPG##Enr**.

**##** = Group number

**nr** = Exercise number (start with 01)



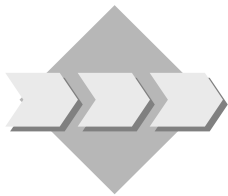
## Unit: The Logical Databases PNP/PNPCE

### Topic: Creating a Customer Report Class



At the end of this exercise, you will be able to:

- Create customer report classes using the IMG
- Assign report classes to customer reports.



The personnel department requires that the selection screens for the customer reports only display the selection fields that are required for the corresponding reports.

- 1-1 Use the IMG to create a customer report class 000000## (where nn = group number) for the logical database PNP. Sorting and matchcodes are permitted and the start and end dates are permitted entries for both the data selection period and the person selection period.

Select the following fields for the first page of the SELECT-OPTIONS:

- Personnel number
- Employment status
- Company code
- Personnel subareas
- Employee group
- Employee subgroup
- Cost center

- 1-2 Assign the new report class to your report.



## Unit: The Logical Databases PNP/PNPCE

### Topic: Creating a List of Employees

```
REPORT zpsol010.
*-- Declaration
TABLES: pernr.
INFOTYPES: 0002.                  "Personal Data
SELECT-OPTIONS: language FOR p0002-sprsl.

*-- Selection screen
INITIALIZATION.
    pnptimed = 'D'.

*-- Processing
GET pernr.
    PROVIDE * FROM p0002 BETWEEN pn-begda AND pn-enddda.
    CHECK language.
    WRITE: / p0002-pernr,
            sy-vline,
            pernr-ename,
            sy-vline,
            p0002-sprsl,
            sy-vline,
            p0002-gbdat.
ENDPROVIDE.
```



## Unit: The Logical Databases PNP/PNPCE

### Topic: Creating a Customer Report Class

#### 1-1 Create customer report class 000000##:

***IMG → Personnel Management → Human Resources Information System → Reporting → Adjusting the Standard Selection Screen → Create Report Categories***

Select *New Entries* and enter the number of your report class and the name. Deactivate the *For Logical Database PNPCE* field.

Select the *Matchcode allowed* and *Sort allowed* indicators and allow the entry of a date interval for the data and person selection period.

Save your entries and create a transport request. Return to the overview screen.

On the overview screen, select your new report class and select *Allowable Selection Criteria* in the dialog structure. Choose *New Entries*. Use the possible entries help (F4) to select the specified selection fields and select the field for output on the first page. Save your entries.

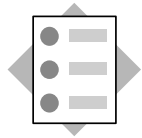
#### 1-2 To assign the customer report category to your own reports, select:

***IMG → Personnel Management → Human Resources Information System → Reporting → Adjusting the Standard Selection Screen → Assign report categories***

Choose *New Entries* and enter your program name and new report class. Save your entries.

## **Contents:**

- **Joining infotypes**
- **Field projection**
- **Combining join and projection**



**At the conclusion of this unit, you will be able to:**

- **Process two or more infotypes in one  
PROVIDE - ENDPROVIDE loop**
- **Determine that only changes to specific fields are  
taken into account when an infotype is processed**
- **Combine join and projection**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration

Data Structures in Personnel Planning

The Logical Databases PNP/PNPCE

The Logical Database PCH



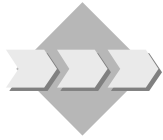
**Join and Projection**

Customer-Specific Infotypes

Specific Commands

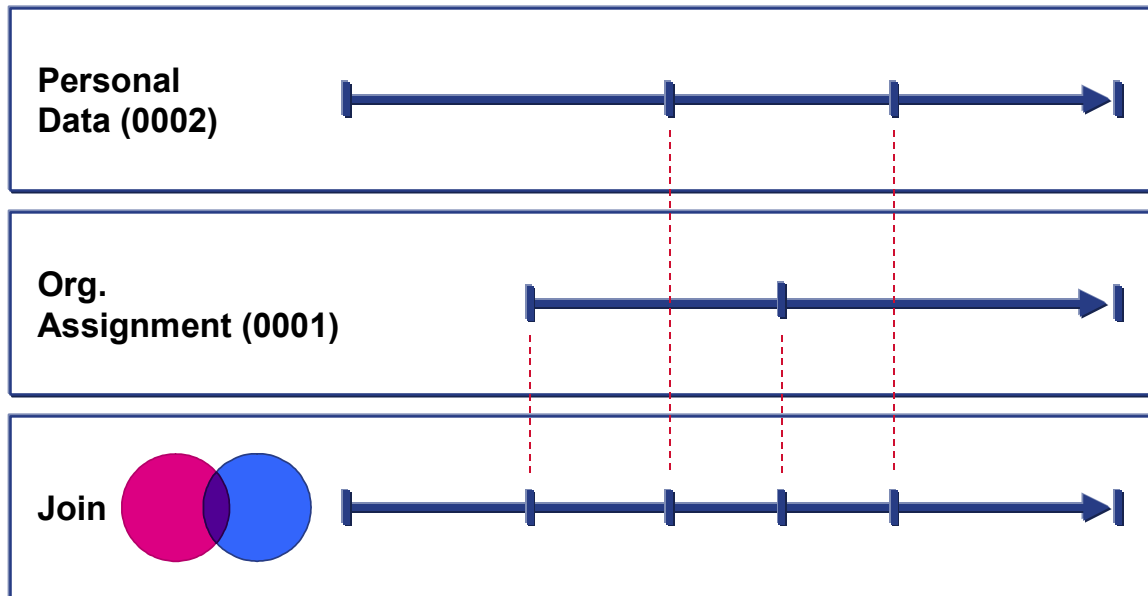
HR ABAP Features

Repetitive Structures and List Display



- **The Personnel Department of your company requires additional reports that combine data from different infotypes.**





© SAP AG 2003

- Evaluations can either be coded for specific infotypes, which means that each infotype is processed in its own PROVIDE loop, or for all infotypes, which means that a single PROVIDE loop retrieves and processes data from two or more infotypes at the same time. The infotypes to be processed are listed as the data source.
- All HR data changes over time. For this reason, time-dependencies must be taken into account when infotype data is linked by a join. HR data is not infinitely valid. It is valid for specific periods only.
- Therefore, the result of a join is one or more validity intervals in which the data of both infotypes is valid.
- If an evaluation is run for key dates, the currently valid data of both infotypes is retrieved in a data record.
- In principle, a join is a logical database operation performed on the time axis. Based on the validity periods of the infotypes to be linked, new periods are created with the data that is valid. In other words, new infotype records are created. This enables you to see the time-based interrelationships between the infotypes in question.

```
TABLES: PERNR.
INFOTYPES: 0001,      "Organizational Assignment
            0002,      "Personal Data
            0006,      "Addresses
            ....

GET PERNR.
  PROVIDE * FROM P0001
          * FROM P0002
          BETWEEN PN-BEGDA AND PN-ENDDA.
  WRITE...

ENDPROVIDE.
```

© SAP AG 2003

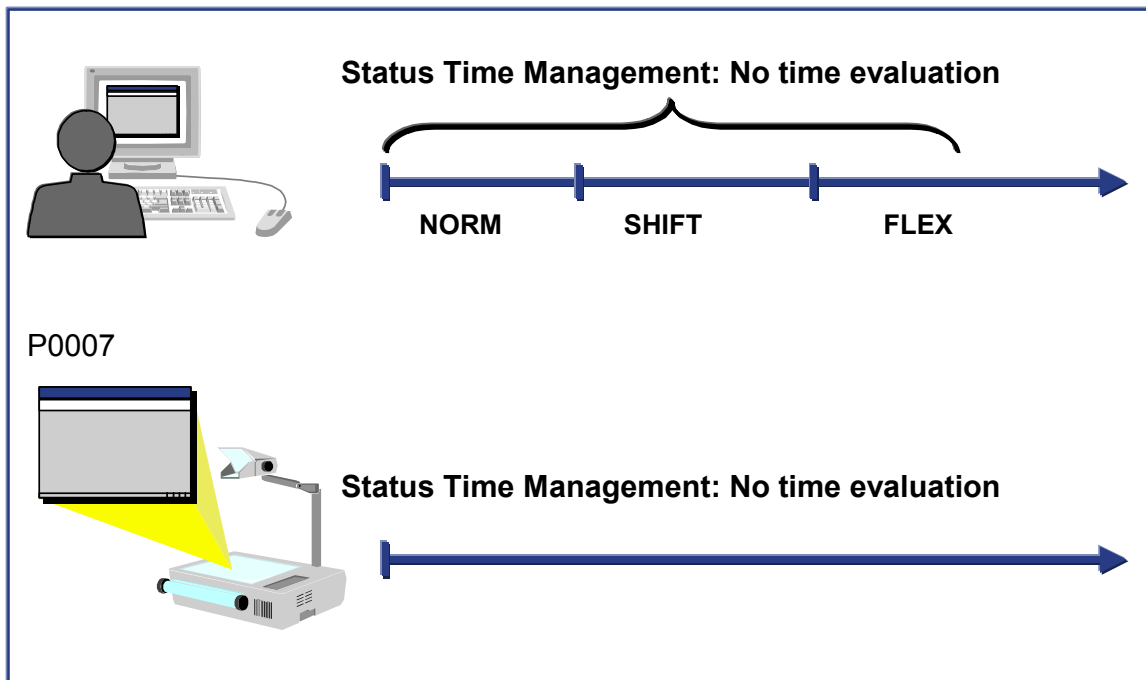
- Any number of infotypes can be linked by a join.
- If more than one record is valid for an infotype at the same time or in the same period (time constraint 3), the data that is valid in the partial interval covered by the join is not meaningful. The resulting restriction for the join is that the data of the infotypes in question must remain unique at all times. This means that joins are only possible for infotypes with **time constraint 1 or 2**.
- The logical validity is stored in the infotype's BEGDA and ENDDA fields.
- Note that changing data in any one of the infotypes linked by a join causes a split in the selection period.

```
TABLES: PERNR.
INFOTYPES: 0001,      "Organizational Assignment
            0002,      "Personal Data
            0006,      "Addresses
            ....

GET PERNR.
  PROVIDE * FROM P0002
          * FROM P0006
            BETWEEN PN-BEGDA AND PN-ENDDA
  WHERE P0006-SUBTY = '1'.
  IF P0006_VALID = 'X'.
    WRITE...
  ENDIF.
ENDPROVIDE.
```

© SAP AG 2003

- If infotypes linked by a join have subtypes, processing must be restricted to one subtype using a WHERE condition, in which a subtype is queried.
- In this example, the first partial interval only contains personal data. The record is not meaningful because the join's task of retrieving data from all of the infotypes in question has not been performed.
- Using the variable Pnnnn\_VALID, the system recognizes that one partial interval only contains incomplete data.
- When the report is run, this variable is created for each Pnnnn infotype included in a join.
- If a partial interval for infotype Pnnnn contains data, its Pnnnn\_VALID variable is filled with X.



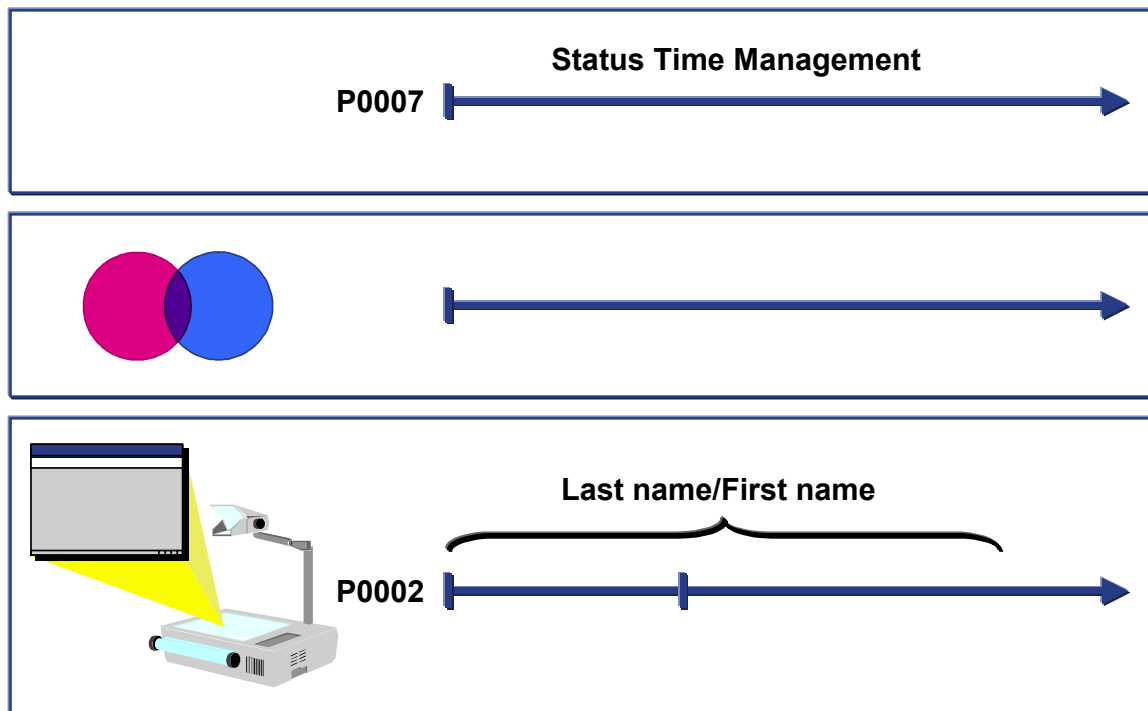
© SAP AG 2003

- Selecting one or more infotype fields for processing is called **projection**.
- Using projection for specific fields enables you to determine that only the contents of these fields, and time-based changes to the contents, are relevant to an evaluation.
- Like joins, a projection is a logical data view concentrated on one or more fields within an infotype.
- Projection is also an operation performed on the time axis. New validity periods are created when the contents of one of the projection fields change. The values stored in fields that are not included in the projection are in an undefined condition, which means they are no longer relevant to output.
- If the contents of a field included in the projection remain unchanged in several data records, the validity periods of these data records are combined.
- The process of combining data records during projection is called **contraction**.

```
TABLES: PERNR.  
INFOTYPES: 0002,      "Personal Data  
            0007,      "Planned Working Time  
            ....  
  
GET PERNR.  
  PROVIDE ZTERF FROM P0007  
            BETWEEN PN-BEGDA AND PN-ENDDA.  
  WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- The \* in the PROVIDE statement, which stands for all of the fields in the infotype, is replaced by the five-digit field name of the infotype included in the projection.
- The \* in a simple PROVIDE means that all of the fields in the infotype are included in the projection.
- Join and projection can be combined in a PROVIDE statement.
- Infotype records are not allowed to overlap for projection. Data must be unique, which means that projection is only allowed for **time constraints 1 and 2**.



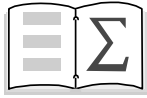
© SAP AG 2003

- These logical views of infotype data - joins and projection - can be combined.
- Data from several infotypes is read at the same time, and new partial periods are created. Since fields are only selected from infotypes if they are relevant to an evaluation, partial periods are combined.
- In this way, validity periods with related contents are combined, and data is retrieved for an evaluation from relevant infotype fields and for an exact period.

```
TABLES: PERNR.  
INFOTYPES: 0002,      "Personal Data  
            0007,      "Planned Working Time  
            ....  
  
GET PERNR.  
  PROVIDE  NACHN  
           VORNA FROM P0002  
           ZTERF FROM P0007  
           BETWEEN PN-BEGDA AND PN-ENDDA.  
  
  WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- Join and projection can be combined in a PROVIDE statement.



**You are now able to:**

- **Process two or more infotypes in one PROVIDE - ENDPROVIDE loop**
- **Determine that only changes to specific fields are taken into account when an infotype is processed**
- **Combine join and projection**





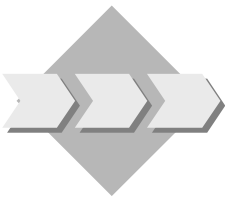
## Unit: Join and Projection

### Topic: Joining Infotypes



At the end of this exercise, you will be able to:

- Process the *Personal Data* infotype (0002) and the *Addresses* infotype (0006) using a join.
- Make the processing of data dependent on the content of the variable Pnnnn\_VALID.



The personnel department of your company requires an overview of all previous and present addresses of employees.

- 1-1 Create a report that documents the period in which the employee lived at which address and under which name. The created list should contain the following information:
  - Personnel number
  - Start date
  - End date
  - Name
  - Place of residence
- 1-1-1 Only process the records from infotype 0006 which have subtype 1.
- 1-1-2 Only print data for the periods for which address information is available.
- 1-1-3 Use a variable with the characteristics of field **P0001-ENAME**, so that the first and last name are only separated by a space. Use the **CONCATENATE** statement.
- 1-1-4 Enter a width of 100 characters for the list (addition **LINE-SIZE** for keyword **REPORT**).  
Limit the length of the name to 30, and the length of the city to 20 characters.  
Separate the resulting field contents in your list by a vertical line (**SY-VLINE**).



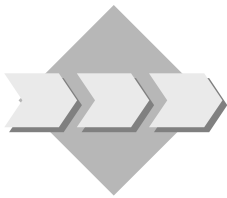
## Unit: Join and Projection

### Topic: Projection to Infotype Fields



At the end of this exercise, you will be able to:

- Project to particular fields by joining two infotypes



Your company's Board of Directors requires a list of employees with the values of the *Status of Time Evaluation* field in the *Planned Working Time* infotype.

- 1-1 Create a report that documents the values of the *Status of Time Evaluation of Employees* field. The list should contain the following information:
  - Personnel number
  - Name
  - Status of time recording
  - Start date
  - End date
- 1-1-1 Project to the fields **P0002-NACHN** and **P0002-VORNA** for infotype 0002. Project to field **P0007-ZTERF** for infotype 0007.
- 1-1-2 Only print data for the periods for which address information is available.
- 1-1-3 The texts for the keys in the *Status of Time Recording* field are contained in the table T555V. Before reading this table, check that the table entry to be read for the corresponding key in question is already in the work area. To do this, use the **CHECK** statement.
- 1-1-4 Enter a width of 100 characters for the list (addition **LINE-SIZE** for keyword **REPORT**).



## Unit: Join and Projection

### Topic: Joining Infotypes

```
REPORT zpsol020 LINE-SIZE 100.
*-- declaration
TABLES: pernr.
INFOTYPES: 0002,           "Personal Data
            0006.           "Address
DATA: name LIKE pernr-ename.

*-- Processing
GET pernr.
  PROVIDE * FROM p0002
    * FROM p0006 BETWEEN pn-begda AND pn-enddda
    WHERE p0006-subty = '1'.
  IF p0006_valid EQ 'X'.
    CONCATENATE p0002-nachn p0002-vorna INTO name
    SEPARATED BY SPACE.
    PERFORM print_data.
  ENDIF.
ENDPROVIDE.

*-----*
*      FORM PRINT_DATA
*-----*
FORM print_data.
  WRITE: / p0002-pernr NO-ZERO,
    sy-vline,
    p0002-begda,
    sy-vline,
    p0002-enddda,
    sy-vline,
    (30) name,
    sy-vline,
    (20) p0006-ort01.
ENDFORM.
```



## Unit: Join and Projection

### Topic: Projection to Infotype Fields

```

REPORT zpsol030 LINE-SIZE 100.
*-- Declaration
TABLES: pernr,
        t555v.                                "Job Titles
INFOTYPES: 0002,                             "Personal Data
           0007.                             "Planned Working Time
DATA: name LIKE pernr-ename.

*-- Processing
GET pernr.
  PROVIDE nachn vorna FROM p0002
        zterf          FROM p0007 BETWEEN pn-begda AND pn-endda.
  IF p0007_valid EQ 'X'.
    CONCATENATE p0002-nachn p0002-vorna INTO name
    INTO name SEPARATED BY space.
    PERFORM re555v USING p0007-zterf.
    WRITE: / pernr-pernr,
            name,
            (20) t555v-ztext,
            p0007-begda,
            p0007-endda.
  ENDIF.
ENDPROVIDE.

*-----*
*      FORM RE555v                                *
*-----*
*      Read Time Management Status                *
*-----*
FORM re555v USING value(tm_status).
  CHECK sy-langu NE t555v-sprsl OR
        tm_status NE t555v-zterf.

  SELECT SINGLE * FROM t555v WHERE sprsl EQ sy-langu
                                AND zterf EQ tm_status.

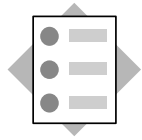
  IF sy-subrc NE 0.
    MOVE space TO t555v.
  ENDIF.
ENDFORM.

```

### **Contents:**

- **Macros**
- **Function modules**

© SAP AG 2003



**At the conclusion of this unit, you will be able to:**

- **Use macros**
- **Use function modules**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration

Data Structures in Personnel Planning

The Logical Databases PNP/PNPCE

The Logical Database PCH

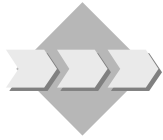
Join and Projection

Customer-Specific Infotypes

 **Specific Commands**

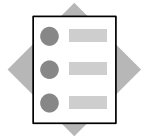
HR ABAP Features

Repetitive Structures and List Display



- **The Executive Board of your company requires lists that display the data valid for an employee in a specified data selection period.**





**At the conclusion of this topic, you will be able to:**

- **Use macros for Personnel Administration logical databases**

© SAP AG 2003

- Like subroutines and function modules, macro modules can be used in HR to modularize source code.
- For documentation on key macros, see the section on programming aids for the logical databases PNP and PAP in SAP Library.

```
TABLES: PERNR.  
INFOTYPES: 0001,      "Organizational Assignment  
            0002,      "Personal Data  
            0006,      "Addresses  
            ....
```

```
GET PERNR.
```

```
  RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGDA PN-ENDDA.
```

```
  WRITE...
```

```
* * * Include program DBPNPMAC.
```

```
DEFINE RP_PROVIDE_FROM_LAST.
```

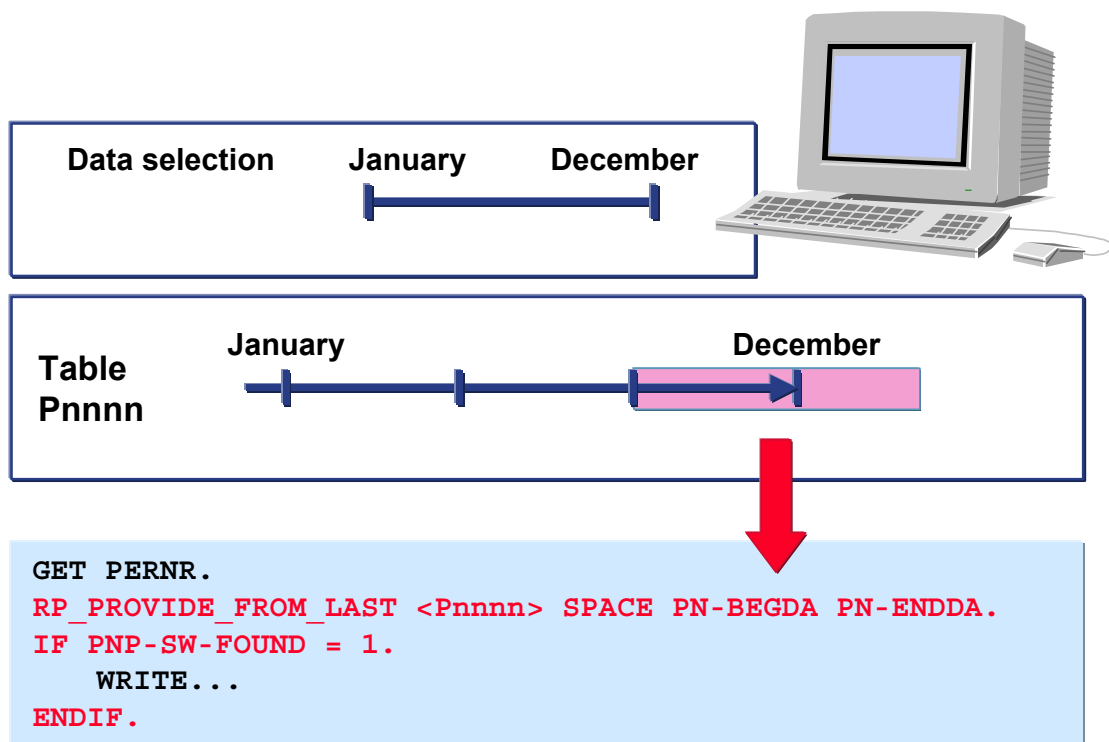
```
  PNP-SW-FOUND = '0'.
```

```
  . . .
```

```
END-OF-DEFINITION.
```

© SAP AG 2003

- Like subroutines and function modules, macro modules can be used to modularize programs. Macro modules are frequently used in HR.
- These macros are defined in the program SAPDBPNP (include DBPNPMAC) with the keyword DEFINE. They can be used in any program that uses the logical database PNP.
- If you want to use these macros in reports that do not use the logical database PNP, you must include program DBPNPMAC with the keyword INCLUDE.
- You can also define your own macros. In accordance with the naming convention, the first two letters stand for the application.
- Some macros are also stored in the table Macros in ABAP Programs (TRMAC)



© SAP AG 2003

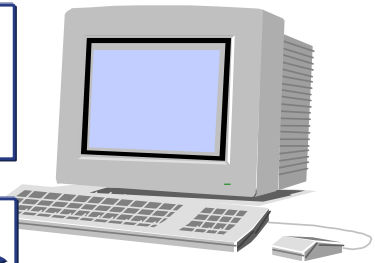
- The `RP_PROVIDE_FROM_LAST` macro retrieves the last valid data record in the data selection period.
- The parameters for `RP_PROVIDE_FROM_LAST` are: Infotype, subtype, start date, and end date. If you do not want to specify a particular subtype, enter `SPACE`.
- As well as processing the last valid data record in the data selection period, you can also process the first valid data record using the `RP_PROVIDE_FROM_FRST` macro.
- The `PNP-SW-FOUND` macro return code has the value 1 if there is a corresponding entry in the infotype table in the specified period. It has the value 0 if no entry was found.

**Data Selection**

**May      December**



**PAxxxx**



**START-OF-SELECTION.**

**RP\_SET\_DATA\_INTERVAL '<Pxxxx>' PN-BEGDA PN-ENDDA.**

**GET PERNR.**



**Table  
Pxxxx**



© SAP AG 2003

- When the START-OF-SELECTION event occurs, the RP\_SET\_DATA\_INTERVAL macro ensures that the only records imported from database table PAxxxx to internal table Pxxxx are those which are valid in the data selection period.
- If the selection specified is to apply to all infotypes, the infotype name in quotation marks must be replaced by **ALL**.

```
TABLES: PERNR, T001P.
INFOTYPES: 0001,
        ....
```

## Personnel Area/Subarea Table

PersArea	Subarea	Groupings ...
...	...	...
CABB	0001	01 01
...	...	...

```
GET PERNR.
RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGDA PN-ENDDA.

RP-READ-T001P P0001-WERKS P0001-BTRTL SPACE.
```

© SAP AG 2003

- It is often necessary to read the groupings stored in the Personnel Area/Subarea table because they are required as keys for other tables. This must take place separately for each personnel number to be processed.
- Each employee's assignment to a personnel area and subarea is stored in infotype 0001.
- The parameters of the macro are:
  - Personnel area
  - Personnel subarea
- If no entry is found, the system reacts as follows:
  - X = Termination of report with the error message "No entry found in table ..."
  - SPACE = SY-SUBRC is set to 4
- Alternatively, if you only want to determine the country indicator, you can use the function module **HR\_COUNTRYGROUPING\_GET**.

```

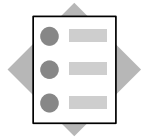
INFOTYPES:  0001,
            0002,
            ...
            2005 MODE N.

GET PERNR.
  RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGDA PN-ENDDA.
  . . . .
  RP_READ_ALL_TIME_ITY PN-BEGDA PN-ENDDA.
  . . . .

  LOOP AT P2005.
    WRITE...
  ENDLOOP.
    
```

© SAP AG 2003

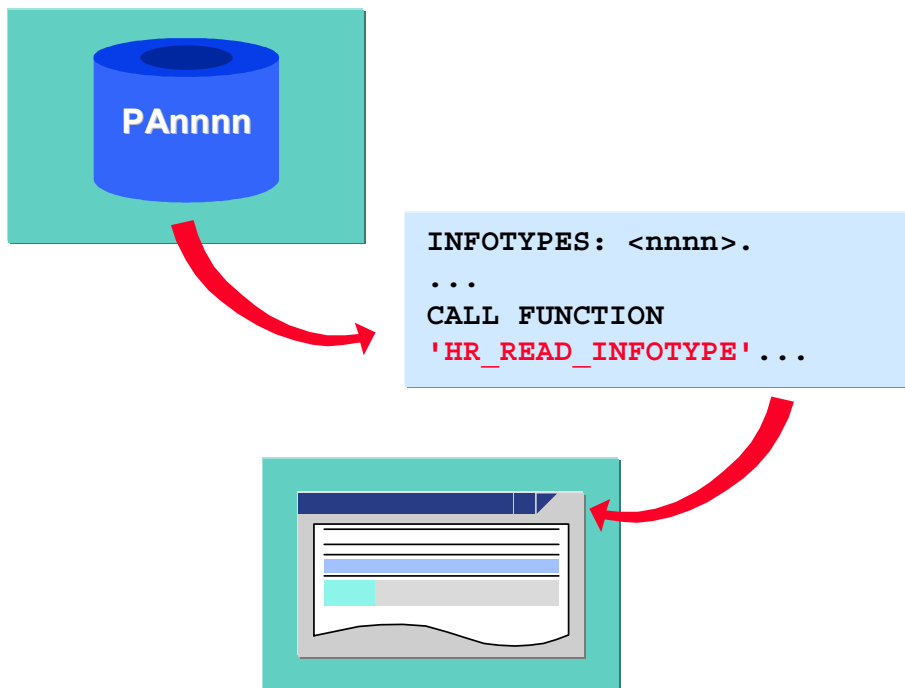
- Due to the large amount of data in HR, the infotypes 2000 – 2999 should not be read when GET PERNR occurs. Therefore, these infotypes are declared with the enhancement **MODE N**.
- As a result, the infotype tables under GET PERNR are not filled. The time infotype tables are filled subsequently using the macro **RP\_READ\_ALL\_TIME\_ITY**, but only for the time interval specified by PN-BEGDA and PN-ENDDA.



**At the conclusion of this topic, you will be able to:**

- **Search for function modules in HR, to use them in your own programs**

- You can use the standard function modules to help you process large quantities of data. Make sure that you check whether allowed entries are not already defined for specific functions in standard function modules.
- Most function modules in HR start with **"RP"** , **"HR"** and **"BAPI"**. In transaction SE37, select the menu *Utilities -> Find*. For example, if you are looking for a function module that determines an employee's hiring date, search for the entries the **"RP\*"** and **"HR\*"** in the selection criterion *Function module* and **"\*Entry\*"** in the *Short description* field.



© SAP AG 2003

- You can also read infotype records for a particular personnel number without using the logical database.
- To do this, use the function module `HR_READ_INFOTYPE`. However, you must ensure that the internal table for the required infotype is declared with the `INFOTYPE` statement.



```

INFOTYPES: 0002.
DATA: return LIKE SY-SUBRC.

CALL FUNCTION 'HR_READ_INFOTYPE'
  EXPORTING
    . . .
    PERNR          = <person>
    INFY           = '0002'
    BEGDA          = <begdat>
    ENDDA          = <enddat>

  IMPORTING
    SUBRC          = return

  TABLES
    INFY_TAB       = P0002

  EXCEPTIONS
    INFY_NOT_FOUND = 1
    OTHERS         = 2.

```

© SAP AG 2003

- The function module reads the HR infotype records for a person (employee or applicant) in accordance with the specified selection criteria. Values are returned in an internal table, the structure of which corresponds to the appropriate infotype table. In the calling program, such tables can be declared with the INFOTYPES statement, for example. An infotype record is selected if its validity period overlaps with the specified period.
- The function module carries out an authorization check.
- The return code can have the following values:
  - 0: The return table contains all the records requested
  - 4: The return table contains records, but is incomplete due to missing authorization
  - 8: The return table is empty as no records were found for the criteria specified
  - 12: The return table is empty due to missing authorization
- **Note:**

You should not use this function module in reports that use the logical database PNP. If you want to read an infotype separately in one of these reports, you can call the subroutine **READ-INFOTYPE** directly in the database program **SAPDBPNP** (PERFORM READ-INFOTYPE(SAPDBPNP)). . .). In this case, you declare the infotypes in the statement INFOTYPES with enhancement **MODE N**.

```
INFOTYPES: 0001.
DATA: hire_date LIKE P0016-EINDT.

CALL FUNCTION 'RP_GET_HIRE_DATE'
  EXPORTING
    PERSNR          = P0001-PERNR
    CHECK_INFOTYPES = '0000'
  *   DATUMSART      = '01'
  *   STATUS2        = '3'
  *   P0016_OPTIONEN = ' '
  IMPORTING
    HIREDATE        = hire_date
  EXCEPTIONS
    OTHERS           = 1.
```

© SAP AG 2003

- You can use this function module to determine an employee's entry date. The following infotypes can be used to determine this date:
  - P0000 Measures
  - P0001 Organizational assignment
  - P0016 Contract components
  - P0041 Date specifications
- The transfer parameter **CHECK\_INFOTYPES** is used to determine which of these infotypes are taken into account. The employment status (STATUS2 parameter) can also be specified for infotype P0000.
- If the infotype P0016 is used, you can decide whether the fields P0016-EINDT and/or P0016-KONDT are used to determine dates. If this transfer parameter is SPACE, both fields are taken into account. Specify the technical field name in the transfer parameter for only one field.
- For infotype P0041, the corresponding date type can be specified in the transfer parameter DATUMSART. The date type '01' is usually used for the technical entry date.

```
DATA: return_struct TYPE BAPIRETURN1,
. . .

CALL FUNCTION 'BAPI_EMPLOYEE_ENQUEUE'
  EXPORTING
    NUMBER           = PERNR-PERNR
  IMPORTING
    RETURN           = return_struct.

IF NOT return_struct IS INITIAL.
  WRITE: / return_struct-TYPE,...
ENDIF.
. . .
* Update Infotype Records
. . .
CALL FUNCTION 'BAPI_EMPLOYEE_DEQUEUE'
  EXPORTING
    NUMBER           = PERNR-PERNR
  IMPORTING
    RETURN           = return_struct.
```

© SAP AG 2003

- Before you change employee data, you must lock the personnel number. You can do this using the function module **BAPI\_EMPLOYEE\_ENQUEUE**.
- Messages are returned in the **RETURN** parameter. If errors occur, this structure contains the following information, for example:
  - Message type (*TYPE* field)
  - Message text (*MESSAGE* field)
 If no errors occur, the structure is blank.
- If an employee's data has been locked, only the user who has locked the data can access the data records linked to the employee. Access is denied to other users. You must therefore unlock the data after it has been changed. You can do this using the function module **BAPI\_EMPLOYEE\_DEQUEUE**.

```

DATA:   return_struct TYPE bapireturn1,
        record_key    TYPE bapipakey.
. . .
GET PERNR.
      LOOP AT P0002 WHERE. . .

          CALL FUNCTION 'HR_INFOTYPE_OPERATION'
            EXPORTING
              INFTY          = '0002'
              NUMBER         = P0002-PERNR
              LOCKINDICATOR  = P0002-SPRPS
              VALIDITYEND    = P0002-ENDDA
              VALIDITYBEGIN  = P0002-BEGDA
              RECORDNUMBER   = P0002-SEQNR
              RECORD         = P0002
              OPERATION       = 'MOD'
              DIALOG_MODE    = '1'
            IMPORTING
              RETURN         = return_struct
              KEY            = record_key.

      ENDLOOP.

```

© SAP AG 2003

- You can use this function module to maintain employee and applicant master data. You can specify one data record. All validations that would normally take place if the infotypes were maintained in dialog mode with the individual maintenance screens are also carried out here. The module returns the same error messages as in dialog mode. This means the error messages for the individual maintenance screens are displayed rather than interpreted. The update is carried out by a 'Call dialog' to the module pool for the infotypes. This means that some restrictions apply to the infotypes processed in that way (see the documentation for the module).
- The following values are amongst those available for the **OPERATION** parameter: MOD (change), COPY (copy), DEL (delete), INS (insert), LIS9 (delimit).
- You can use the **DIALOG\_MODE** parameter to specify whether the action is generally run in the background or whether it only runs in the background until an error occurs, or whether the changes are generally carried out in dialog mode. Possible values:
  - '0' The changes are generally processed in the background. If an error occurs (with an E or A message), the entire step is terminated and the module returns the corresponding error message in the 'RETURN' structure.
  - '1' The changes are processed in the background. If an error occurs, the system switches to dialog mode so that the user can correct the entries.
  - '2' The changes are processed in dialog mode.

```
PARAMETERS:natio_o LIKE P0002-NATIO DEFAULT 'DE',
            natio_n LIKE P0002-NATIO DEFAULT 'D'.

* Update Infotype-records

UPDATE PA0002
      SET NATIO      = natio_n
      WHERE PERNR    = P0002-PERNR
      AND NATIO      = natio_o.

IF SY-SUBRC = 0.
  WRITE: 'Modified records', SY-DBCNT.
ENDIF.
```

© SAP AG 2003

- If you cannot use the function module HR\_INFOTYPE\_OPERATION, you can use the UPDATE statement. Note that the system does not check whether the new field contents are correct when a direct database update takes place.
- In this example, the field P0002-NATIO (nationality) is updated directly in DB table PA0002. The field contents are changed from "DE" to "D". The incorrect value "D" in the field PA0002-NATION then triggers an error message when the infotype record is changed or copied in dialog mode.
- The SY-DBCNT system field contains the number of changed records.
- Authorization checks are not supported by the UPDATE statement and should be carried out at the program level.

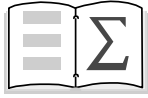
```

CALL FUNCTION 'HR_CHECK_AUTHORITY_INFTY'
  EXPORTING
    TCLAS      = 'A'
    PERNR      = <person>
    INFTY      = '0006'
    SUBTY      = '*'
    BEGDA      = <begdat>
    ENDDA      = <enddat>
    LEVEL      = 'R'
  EXCEPTIONS
    NO_AUTHORIZATION = 1
    INTERNAL_ERROR   = 2
    OTHERS           = 3.

IF SY-SUBRC = 0.
  SELECT * FROM pa0006
        INTO TABLE...
ENDIF.
    
```

© SAP AG 2003

- Authorization checks are not supported by the SELECT statement and must therefore be executed at the program level.
- In this case, you must use the function module **HR\_CHECK\_AUTHORITY\_INFTY** to check whether the user has the required authorization for data and persons.
- If the infotype does not have any subtypes, you can use the SPACE value with the SUBTY parameter.
- SAP recommends that you use the logical database PNP as it automatically runs authorization checks.



**You are now able to:**

- **Use macros**
- **Use function modules**







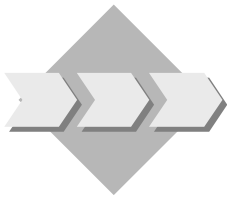
## Unit: Specific Commands

### Topic: Macros



At the end of this exercise, you will be able to:

- Use the RP\_SET\_DATA\_INTERVAL macro to read only the records valid in the evaluation period.
- Use the RP\_PROVIDE\_FROM\_LAST macro to retrieve the last valid record in the data selection period.
- Use the RP-READ-T001P macro to read the Personnel Area/Personnel Subarea table.



The Executive Board of your company requires a list of employees with recurring payments and deductions.

- 1-1 Create a report that will generate a list of the recurring payments and deductions for each employee.

The list should contain the following information for each employee:

- Personnel number
- Name
- Personnel area
- Personnel subarea text

In the following lines, print:

- Wage type
- Wage type text
- Amount
- Start date
- End date

- 1-1-1 Use the **RP\_SET\_DATA\_INTERVAL** macro to specify that only records of infotype 0014 are read for the evaluation period. Only employees who have recurring payments or deductions in this period should appear on the list. If employees do not have a record for infotype 0014, the internal table P0014 is empty when the event GET PERNR occurs.
- 1-1-2 To place the last infotype 0001 record valid in the data selection period in the header of the internal table P0001, use the macro **RP\_PROVIDE\_FROM\_LAST**.
- 1-1-3 Use the **RP-READ-T001P** macro to read the table T001P (Personnel Area/Subarea). The country indicator is assigned to each personnel area/subarea in the field **T001P-MOLGA**. The country indicator is part of the key for table T512T (Wage Type Texts). Use the personnel area text in field **T001P-BTEXT**.
- 1-1-4 Read the corresponding wage type texts for the wage types for infotype 0014 from table T512T. Before reading table T512T, check that the table entry to be read is already in the work area. To do this, use the **CHECK** statement. Use a maximum of 20 characters for the name.



## Unit: Specific Commands

### Topic: Macros

```
REPORT zpsol040.
*-- Declaration
TABLES: pernr,
        t512t,           "Wage type text
        t001p.           "Plant section
INFOTYPES: 0001,         "Organ. assignment
            0014.         "Recur.Payments/Deds.

Start-of-selection.
    rp_set_data_interval 'P0014' pn-begda pn-enddda.

*-- Processing
GET pernr.
    CHECK lines( p0014 ) GT 0.
    rp_provide_from_last p0001 space pn-begda pn-enddda.
    rp-read-t001p p0001-werks p0001-btrtl space.
    SKIP 2.
    WRITE: / pernr-pernr,
            (20) pernr-ename,
            p0001-werks,
            t001p-btext.

    LOOP AT p0014 WHERE enddda GE pn-begda
            AND begda LE pn-enddda.
        CHECK p0014-betrg NE 0.
        PERFORM print USING
            p0014-lgart
            p0014-betrg
            p0014-begda
            p0014-enddda.
    ENDLOOP.
```

```

*-----*
*          FORM PRINT                      *
*-----*
*          Print List                      *
*-----*

```

```

FORM print USING
      value(wtype)
      value(amount)
      value(begda)
      value(endda).

```

```

PERFORM re512t USING t001p-molga wtype.
WRITE: / wtype UNDER pernr-ename,
        t512t-lgtxt,
        amount,
        begda,
        endda.

```

```

ENDFORM.                                "print

```

```

*-----*
*          FORM RE512T                      *
*-----*
*          Read Wagetyp Texts              *
*-----*

```

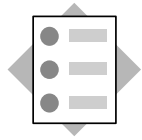
```

FORM re512t USING value(country_grouping)
                  value(wtype).
CHECK t512t-sprsl NE sy-langu
      OR t512t-molga NE country_grouping
      OR t512t-lgart NE wtype.
SELECT SINGLE * FROM t512t
      WHERE sprsl EQ sy-langu
      AND   molga EQ country_grouping
      AND   lgart EQ wtype.
IF sy-subrc NE 0.
  CLEAR t512t.
ENDIF.
ENDFORM.

```

## **Contents:**

- **Evaluating infotypes with repetitive structures**
- **Displaying lists with the ABAP List Viewer**



**At the conclusion of this unit, you will be able to:**

- **Evaluate infotypes with repetitive structures in loop processing**
- **Display lists using the ABAP List Viewer**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration

Data Structures in Personnel Planning

The Logical Databases PNP/PNPCE

The Logical Database PCH

Join and Projection

Customer-Specific Infotypes

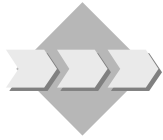
Specific Commands

HR ABAP Features



**Repetitive Structures and List Display**

© SAP AG 2003



- Your company's Personnel Department would like to edit an overview of all employees with data in the *Date Specifications* infotype.



Date specifications 0041

Date type		Date
01	Technical entry date	January 01, 1994
40	First working day	January 16, 1994
—	_____	_____
—	_____	_____
—	_____	_____
—	_____	_____

Dictionary P0041

...	
DAR01	Date type
DAT01	Date per date type
DAR02	...
DAT02	...
...	

0005    Leave Entitlement

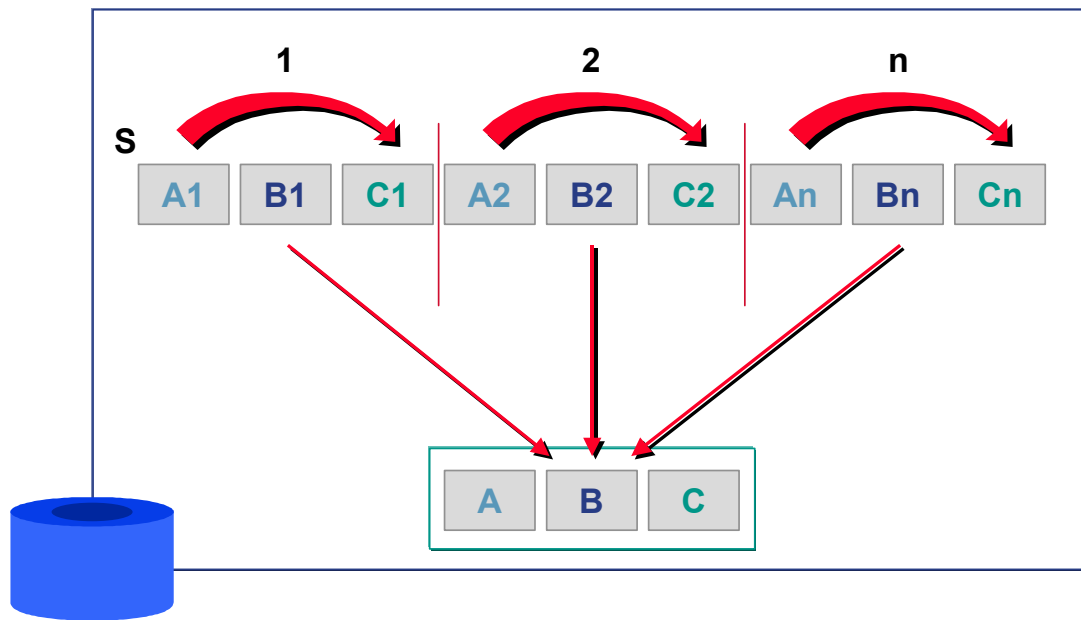
0008    Basic Pay

0025    Appraisals

0027    Cost Distribution

0041    Date Specifications

0052    Standard Wage  
         Maintenance



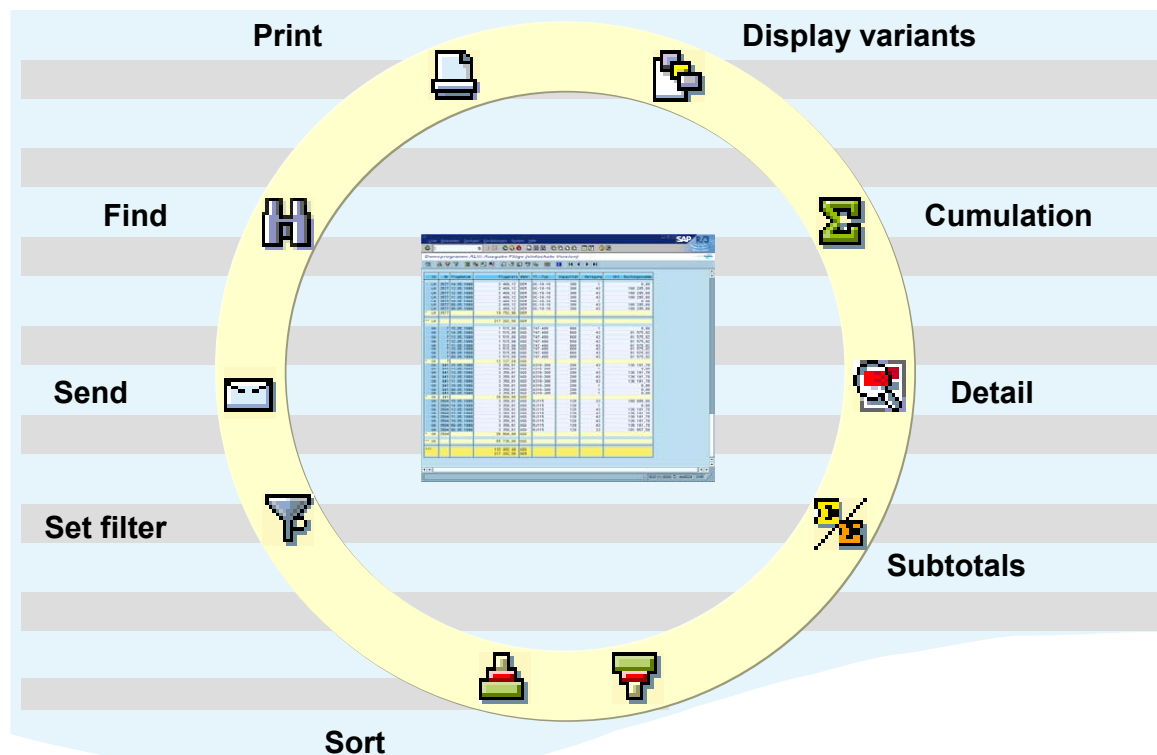
© SAP AG 2003

```
INFOTYPES: 0041,      "Date Specifications
            0008.      "Basic Pay
```

```
DATA:  BEGIN OF specification,
        dar LIKE P0041-DAR01,
        dat LIKE P0041-DAT01,
      END OF specification.
```

```
DATA:  BEGIN OF wagetypes,
        lga LIKE P0008-LGA01,
        bet LIKE P0008-BET01,
        anz LIKE P0008-ANZ01,
        ein LIKE P0008-EIN01,
        opk LIKE P0008-OPK01,
      END OF wagetypes.
```

```
GET PERNR.  
  RP_PROVIDE_FROM_LAST P0041 SPACE PN-BEGDA PN-ENDDA.  
  
DO 12 TIMES VARYING specification  
  FROM P0041-DAR01          "Starting point  
  NEXT P0041-DAR02.         "Increment  
  IF specification-dar IS INITIAL.  
    EXIT.  
  ELSE.  
    WRITE: / specification-dar ....  
  ENDIF.  
ENDDO.
```



© SAP AG 2003

```
. . .
TYPE-POOLS: SLIS.
. . .
DATA:    alv_fieldcat TYPE slis_t_fieldcat_alv,
         alv_layout TYPE slis_layout_alv.
. . .

TYPES:   BEGIN OF data_struct,
         pernr LIKE PERNR-PERNR,
         ename LIKE P0001-ENAME,
         . . .
         dar   LIKE P0041-DAR01,
         . . .
         END OF data_struct.

DATA:    data_tab TYPE TABLE OF data_struct,
         data_tab_wa TYPE data_struct.

DATA:    stru_disvar TYPE disvariant.
```

© SAP AG 2003

```
FORM fieldcat_init USING p_fieldcat
                        TYPE slis_t_fieldcat_alv.
  DATA: ls_fieldcat TYPE slis_fieldcat_alv.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname      = 'PERNR'.
  ls_fieldcat-ref_tabname   = 'PERNR'.
  ls_fieldcat-key           = 'X'.
  APPEND ls_fieldcat TO p_fieldcat.

  . . .
  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname      = 'DAR'.
  ls_fieldcat-ref_fieldname  = 'DAR01'.
  ls_fieldcat-ref_tabname    = 'P0041'.
  APPEND ls_fieldcat TO p_fieldcat.

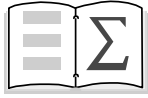
  . . .
ENDFORM.
```

```
END-OF-SELECTION.
  PERFORM fieldcat_init USING alv_fieldcat.
  alv_layout-colwidth_optimize = 'X'.
  . . .

  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
      I_CALLBACK_PROGRAM      = 'Reportname'
    . . .
      I_GRID_TITLE            = 'Title'
    . . .
      IS_LAYOUT               = alv_layout
      IT_FIELDCAT             = alv_fieldcat
      I_SAVE                  = 'A'
      IS_VARIANT              = stru_disvar
    . . .
  TABLES
    T_OUTTAB                 = data_tab
  EXCEPTIONS
    PROGRAM_ERROR            = 1
    OTHERS                   = 2.
```

© SAP AG 2003





**You are now able to:**

- **Evaluate infotypes with repetitive structures in loop processing**
- **Display lists using the ABAP List Viewer**





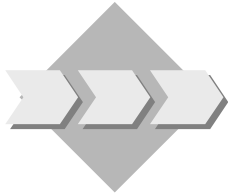
## Unit: Repetitive Structures and List Display

### Topic: Date Specifications Infotype



At the end of this exercise, you will be able to:

- Evaluate the repetitive structures in the *Date Specifications* infotype
- Display the list using the function module `REUSE_ALV_GRID_DISPLAY`



The Personnel Department requires a report that creates an overview of employees with records for the *Date Specifications* infotype. This overview can then be used for further processing.

1-1 Create a report to format the following employee data using the ABAP List Viewer:

- Personnel number
- Name
- Entry date
- Date type
- Text for date type
- Date

The report should also enable you to restrict the list to one or more date types, and only employees with records for the *Date Specifications* infotype (0041) should appear in the overview.

The user can store display variants.

1-1-1 To place the last records for infotypes 0001 and 0041 that are valid in the selection period in the header of the internal tables P0001 and P0041, use the macro

**RP\_PROVIDE\_FROM\_LAST.**

For infotype 0041, query the return code PNP-SW-FOUND for the macro. If the employee has no valid record in the data selection period, the return value is 0. End further processing of the personnel numbers with the **REJECT** statement.

1-1-2 Read the texts for the date types from table **T548T**.

1-1-3 To determine the hiring date from the *Actions* infotype, use the function module **RP\_GET\_HIRE\_DATE**.

- 1-1-4 Set the layout parameters **COLWIDTH\_OPTIMIZE** and **ZEBRA** to the value "X".
- 1-1-5 You can use the following includes:  
**ZPSOL050\_DATA\_STRUCTURES** contains the required data definitions.  
**ZPSOL050\_FILL\_FIELDCAT** contains the subroutines for filling the field catalog.



## Unit: Repetitive Structures and List Display

### Topic: Date Specifications Infotype

REPORT zpsol050.

\*-- declaration

TABLES: pernr, t548t.	"Date Types
INFOTYPES: 0001,	"Org.Assignment
0041.	"Date Specifications

```

*$*$$- Start of INCLUDE ZPSOL050_DATA_STRUCTURES          --$*$*
*&-----*
*& Include                ZPSOL050_DATA_STRUCTURES          *
*&-----*

```

TYPE-POOLS: slis.

DATA: BEGIN OF dtype,	"Work area for IT0041
dar LIKE p0041-dar01,	
dat LIKE p0041-dat01,	
END OF dtype.	

TYPES: BEGIN OF data_struct,	"Data table structure
pernr LIKE pernr-pernr,	
ename LIKE pernr-ename,	
eindt LIKE p0016-eindt,	
dar LIKE p0041-dar01,	
dtext LIKE t548t-dtext,	
dat LIKE p0041-dat01,	
END OF data_struct.	

DATA: data\_tab TYPE TABLE OF data\_struct,  
      data\_tab\_wa TYPE data\_struct.

DATA: alv\_fieldcat TYPE slis\_t\_fieldcat\_alv,  
      alv\_layout TYPE slis\_layout\_alv.

DATA: filled\_lines LIKE sy-index,  
      stru\_disvar TYPE disvariant.

```

*$*$$$--- End of INCLUDE ZPSOL050_DATA_STRUCTURES          --$*$*

```

SELECT-OPTIONS: datatype FOR p0041-dar01 DEFAULT '01'.

```

*-- Processing
GET pernr.
  CHECK lines( p0041 ) GT 0.
  rp_provide_from_last p0001 space pn-begda pn-enddda.
  rp_provide_from_last p0041 space pn-begda pn-enddda.
  IF pnp-sw-found EQ 0.
    REJECT.
  ELSE.
    DO 12 TIMES VARYING dtype FROM p0041-dar01
                                NEXT p0041-dar02.
    IF dtype-dar IS INITIAL.
      EXIT.
    ELSE.
      IF dtype-dar IN datatype.
        CLEAR data_tab_wa.
        MOVE-CORRESPONDING pernr TO data_tab_wa.
        MOVE-CORRESPONDING dtype TO data_tab_wa.
        PERFORM read_hiredate.
        PERFORM re548t USING sy-langu dtype-dar.
        MOVE t548t-dtext TO data_tab_wa-dtext.
        APPEND data_tab_wa TO data_tab.
      ENDIF.
    ENDIF.
  ENDDO.
ENDIF.

```

```

END-OF-SELECTION.
  PERFORM fieldcat_init USING alv_fieldcat.
  alv_layout-colwidth_optimize = 'X'.
  alv_layout-zebra              = 'X'.

  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
      *      I_INTERFACE_CHECK           = ' '
      *      I_BUFFER_ACTIVE             = ' '
      *      i_callback_program          = 'ZPSOL060'
      *      I_CALLBACK_PF_STATUS_SET    = ' '
      *      I_CALLBACK_USER_COMMAND     = ' '
      *      I_CALLBACK_TOP_OF_PAGE      = ' '
      *      I_CALLBACK_HTML_TOP_OF_PAGE = ' '
      *      I_CALLBACK_HTML_END_OF_LIST = ' '
      *      I_STRUCTURE_NAME            =
      *      I_BACKGROUND_ID             = ' '
      *      I_GRID_TITLE                 = 'Date Specifications'
      *      I_GRID_SETTINGS              =
      *          is_layout                 = alv_layout
      *          it_fieldcat               = alv_fieldcat
      *      IT_EXCLUDING                 =
      *      IT_SPECIAL_GROUPS            =
      *      IT_SORT                      =
      *      IT_FILTER                    =
      *      IS_SEL_HIDE                  =
      *      I_DEFAULT                    = 'X'
      *      I_SAVE                       = 'A'
      *      IS_VARIANT                   = stru_disvar
      *      IT_EVENTS                    =
      *      IT_EVENT_EXIT                =
      *      IS_PRINT                     =
      *      IS_REPREP_ID                 =
      *      I_SCREEN_START_COLUMN        = 0
      *      I_SCREEN_START_LINE          = 0
      *      I_SCREEN_END_COLUMN          = 0
      *      I_SCREEN_END_LINE            = 0
      *      IT_ALV_GRAPHICS               =
      *      IT_ADD_FIELDCAT               =
      *      IT_HYPERLINK                  =
      *      IMPORTING
      *          E_EXIT_CAUSED_BY_CALLER   =
      *          ES_EXIT_CAUSED_BY_USER    =

  TABLES
      t_outtab                          = data_tab

  EXCEPTIONS
      program_error                     = 1
      OTHERS                             = 2.

  IF sy-subrc = 1.
      WRITE: 'Program error' (002).
  ENDIF.

```

```

*-----*
*          FORM READ_HIREDATE
*-----*
*          Retrieving hiredate from infotype 0000
*-----*
FORM read_hiredate.
  CALL FUNCTION 'RP_GET_HIRE_DATE'
    EXPORTING
      persnr          = p0001-pernr
      check_infotypes = '0000'
    IMPORTING
      hiredate        = data_tab_wa-eindt
    EXCEPTIONS
      OTHERS          = 1.
ENDFORM.

```

```

*-----*
*          FORM RE548T
*          Read date specification texts
*-----*
VALUE (LANGUAGE)
* -->  VALUE (DTYPE)
*-----*
FORM re548t USING value(language)
                  value(dtype).
  CHECK t548t-sprsl NE language
    OR t548t-datar NE dtype.
  SELECT SINGLE * FROM t548t
    WHERE sprsl EQ language
    AND   datar EQ dtype.
  IF sy-subrc NE 0.
    CLEAR t548t.
  ENDIF.
ENDFORM.

```



```

*$*$$- Start of INCLUDE ZPSOL050_FILL_FIELDCAT                --$*$*
*&-----*
*&   Include                ZPSOL050_FILL_FIELDCAT            *
*&-----*

*-----*
*           FORM FIELDCAT_INIT
*-----*
FORM fieldcat_init USING p_fieldcat TYPE slis_t_fieldcat_alv.
  DATA: ls_fieldcat TYPE slis_fieldcat_alv.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'PERNR'.
  ls_fieldcat-ref_tabname = 'PERNR'.
  ls_fieldcat-key       = 'X'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'ENAME'.
  ls_fieldcat-ref_tabname = 'PERNR'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'EINDT'.
  ls_fieldcat-ref_tabname = 'P0016'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'DAR'.
  ls_fieldcat-ref_fieldname = 'DAR01'.
  ls_fieldcat-ref_tabname = 'P0041'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'DTEXT'.
  ls_fieldcat-ref_tabname = 'T548T'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'DAT'.
  ls_fieldcat-ref_fieldname = 'DAT01'.
  ls_fieldcat-ref_tabname = 'P0041'.
  APPEND ls_fieldcat TO p_fieldcat.
ENDFORM.
*$*$$--- End of INCLUDE ZPSOL050_FILL_FIELDCAT                --$*$*

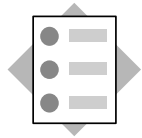
```



### **Contents:**

- **Exporting/importing database tables in HR**
- **Exporting/importing cluster data structures**
- **Exporting/importing with macros using a buffer**
- **Cluster authorization**
- **Data structure for payroll results**
- **Reading the cluster directory**
- **Processing the cluster directory**
- **Processing payroll results**
- **Utilities**

© SAP AG 2003



**At the conclusion of this unit, you will be able to:**

- **Import cluster data structures**
- **Process payroll results using function modules**
- **Use the logical database for payroll results**

Course Overview

**Cluster Database Tables in HR**

Data Structures in  
Personnel Administration

Data Structures in  
Personnel Planning

The Logical Databases  
PNP/PNPCE

The Logical Database PCH

Join and Projection

Customer-Specific Infotypes

Specific Commands

HR ABAP Features

Repetitive Structures and  
List Display

© SAP AG 2003

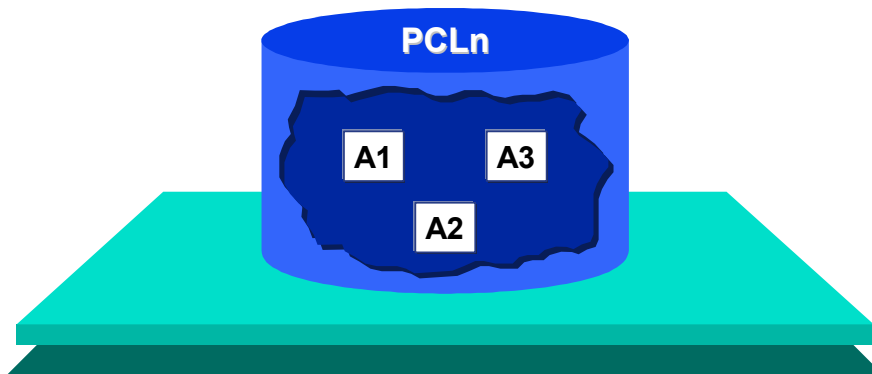


- The different departments in your company want report lists to include the explanatory texts entered when the infotypes are maintained.
- The payroll department would like to use a special report to evaluate the payroll results.



© SAP AG 2003

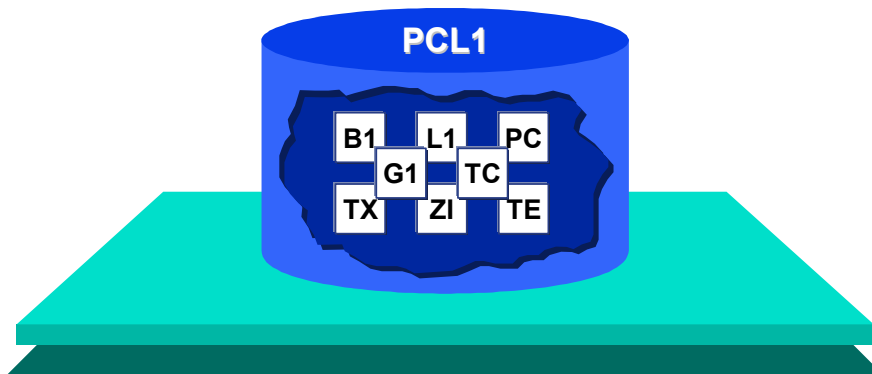
- The PAnnnn database tables contain all HR data sorted by infotype. They constitute the database for infotype entry screens and are evaluated by the HR logical database.
- The PCL1, PCL2, and PCL3 database tables constitute either the database for subsequent programs, such as accounting runs or evaluations, or the database for subareas within Human Resources, such as travel expenses and recruitment.
- The PCLn database tables are a type of import/export database table.



© SAP AG 2003

- Database tables of type PCLn are divided into subareas known as data clusters.
- Data clusters can be identified by their two-character IDs.
- The respective subareas within Human Resources work on their own cluster.
- A separate data key is defined for each subarea.
- You can use the transaction **PECLUSTER** to call a view with information on the existing clusters and the database tables in which they are stored.

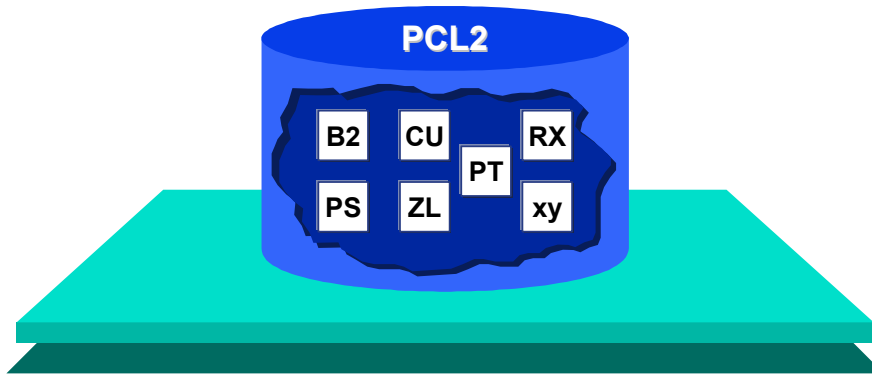




© SAP AG 2003

■ The following are some of the data areas in the database table PCL1:

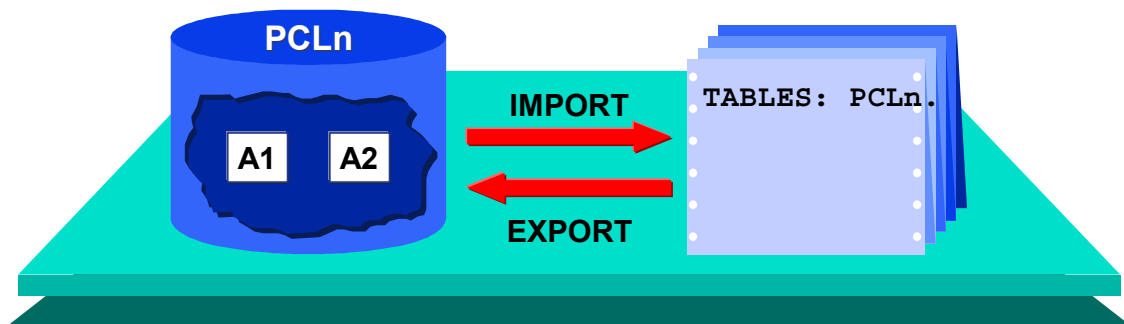
- B1 Time events/PDC
- G1 Group incentive wages
- L1 Individual incentive wages
- PC Personal calendar
- TE Travel expenses/Accounting results
- TC Travel expenses/Credit card data
- TX Infotype texts
- ZI Interface PDC -> Cost Accounting/Materials Management



© SAP AG 2003

■ Database table PCL2 contains the following data areas:

- B2 Time accounting results
- CU Cluster directory
- CA Cluster directory for archived payroll results
- PS Generated schema
- PT Texts for generated schema
- RX Payroll results/international
- Xy Payroll results/country-specific, whereby xy represents the relation ID. This either consists of Rn (n = HR country indicator), or of the ISO code from table T500L.
- ZL Personal work schedule



© SAP AG 2003

- The PCLn cluster database tables are managed by the ABAP commands IMPORT and EXPORT.
- You can use these commands to store any data object - such as fields, structures or internal tables - on the database, or to read them from the database.
- Data is read and written using a unique key.

Field name	KEY	Length	Text
<b>CLIENT</b>	<b>X</b>	<b>3</b>	<b>Client</b>
<b>RELID</b>	<b>X</b>	<b>2</b>	<b>Relation ID</b>
<b>SRTFD</b>	<b>X</b>	<b>40</b>	<b>Work area key</b>
<b>SRTF2</b>	<b>X</b>	<b>10</b>	<b>Sort field for duplicate key</b>

**Example of International Payroll Results:**

<b>RELID</b>	<b>SRTFD</b>
RX	00001911 00001
RX	00001911 00002
RX	00001911 00003
...	

© SAP AG 2003

- The structure of PCLn database tables provides a basic structure for individual data areas.
- The name of each data area must include a two-character cluster name or relation ID.
- A key structure must also be defined; 40 bytes are available in field SRTFD for this purpose.
- In the international payroll results, for example, the field **RELID** contains the cluster name RX and the field **SRTFD** contains the eight-digit personnel number as well as a five-digit sequence number for each individual payroll result for a payroll period.
- The RELID field must contain the cluster identifier and the SRTFD field must contain the cluster key to enable the import of a record from a PCLn database table. The system fills the **SRTF2** field.
- The **CLIENT** field must only be filled if cluster data is to be imported from another client.

```
* KEY-Definition
DATA:   BEGIN OF xy-key,
        FIELD1,
        FIELDn,
        END OF xy-key.

* Definition of Data Objects
DATA:   BEGIN OF table1 OCCURS 10,
        COLUMN1,
        COLUMNn,
        END OF table1.
DATA:   BEGIN OF table2 OCCURS 10, . . .
```

© SAP AG 1999

- The data definitions of a work area are stored in separate programs using a fixed naming convention.
- They are defined as INCLUDE programs. Their names follow convention RPCnxyz0, where.
  - n = 1 or 2 for PCL1 or PCL2
  - xy = Cluster, for example, RX
  - z = 0 for international clusters or country indicator from table T500L for country-specific clusters.
  - Example: The program RPC1TX00 contains the data definition of cluster TX. In this cluster, the database table PCL1 contains the texts stored for infotypes.
- The key structure of the cluster is stored in a field string xy-KEY where the personnel number is the first element.
- The data objects (field strings and tables) are named individually per cluster.

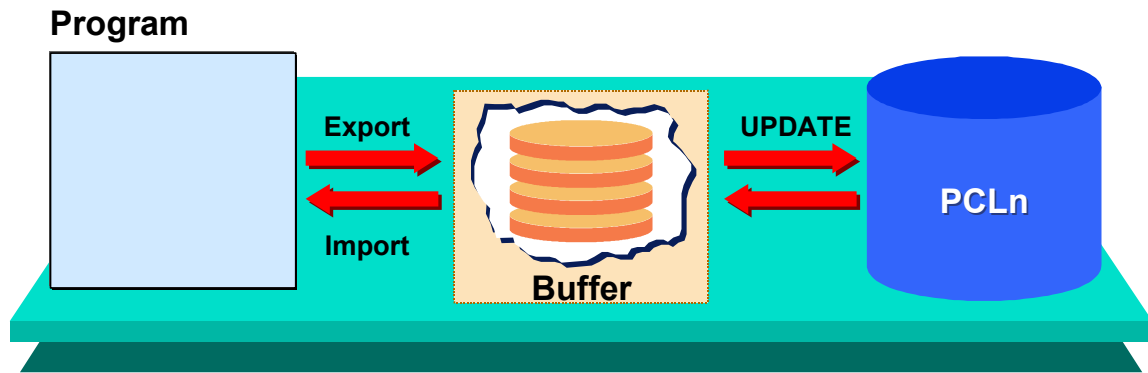
```

TABLES: PCLn,...           "Import/export table
INCLUDE: RPCnxyz0.         "Cluster definition . .

* Fill Cluster-KEY
xy-KEY-FIELD1 = <VALUE>,
. . .
* Import Record
RP-IMP-Cn-xy.
* Display data object
IF SY-SUBRC EQ 0.
    LOOP AT table1.
        WRITE. . .
    ENDLOOP.
ENDIF.
. . .
    
```

© SAP AG 2003

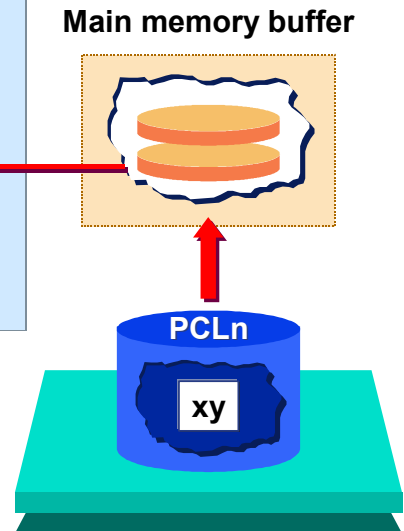
- To ensure consistency when data is exported and imported, the IMPORT/EXPORT commands are defined as macros.
- It is possible to import only a portion of the data objects in a cluster.
- The naming conventions for the macros are RP-IMP-Cn-xy and RP-EXP-Cn-xy, where n is the file name and xy the cluster name.
- If a record is read successfully, the return code is 0. If a record is not read successfully, the return code is 4.
- The macro **RP-IMP-C1-TX** for importing infotype texts to cluster TX is contained in the table Macros in ABAP Programs.
- The macros use routines that carry out two tasks:
  - Data buffering
  - Cluster authorization check



© SAP AG 2003

- To minimize the number of times that the database is accessed, import and export data is buffered in the main memory.
- If a test run is performed, the database is not updated. However, the payroll results of the previous period form the basis of the calculation used to determine the results of the subsequent period. For this reason, the results of a live payroll run and the results of a test run differ if test runs are performed for several periods.
- Using the buffer enables you to access the required results from the previous period.

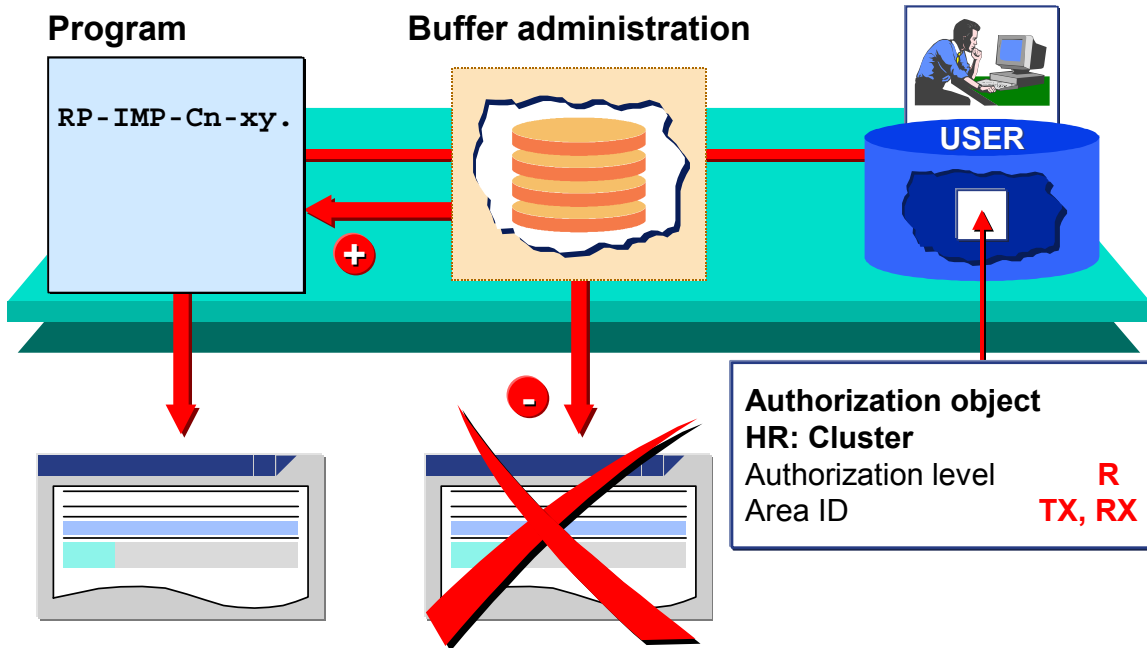
```
* Puffer definition
INCLUDE: RPPPX00.
DATA: BEGIN OF COMMON PART buffer.
      INCLUDE RPPPX10.
DATA: END OF COMMON PART buffer.
* Fill cluster KEY
. . .
* Data import
RP-IMP-Cn-xy.
. . .
* Buffer administration routines
INCLUDE: RPPPXM00.
```



© SAP AG 2003

- If data is imported using macros, the data records are not read directly from table PCLn. Instead, the buffer directory is checked to determine whether the main memory already contains a record with the same key. If this is not the case, the record is read from PCLn to the buffer, and retrieved from the buffer by the report.
- The general data definitions for buffering are contained in the includes RPPPX00 and RPPPX10. The subroutines called by the macro must be incorporated using the include RPPPXM00.
- If data is read using a buffer, the system checks the cluster authorization. The standard import programs follow the RPCLSTxy naming convention, where xy = cluster name.





© SAP AG 2003

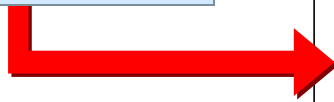
- If the IMPORT statement is used with a buffer, the buffer administration routines check the cluster authorizations (HR authorization object: Cluster).
- The authorization level field enables you to specify the type of access that is allowed:  
**R** provides read access, **U** provides read and write access, **S** allows data to be exported to the PCLx buffer without changing the database (simulated payroll runs).
- The *Area ID* field enables you to list the clusters for which authorization exists.



**At the conclusion of this topic, you will be able to:**

- **Use function modules to read and process payroll results**

**REPORT**  
**H99\_DISPLAY\_PAYRESULT**



## Payroll result

**Kathy Calc**      **January 19xx**

<b>WT01</b>	<b>Standard salary</b>	<b>5000 UNI</b>
<b>WT02</b>	<b>Bonus</b>	<b>300 UNI</b>
<b>WT03</b>	<b>Overtime</b>	<b>200 UNI</b>
<b>/101</b>	<b>Total gross amt</b>	<b>5500 UNI</b>

....

© SAP AG 1999

- With this report, you can display the payroll results for any country. The indicator behind each name shows whether results exist in the selected period.

Green:    Results exist

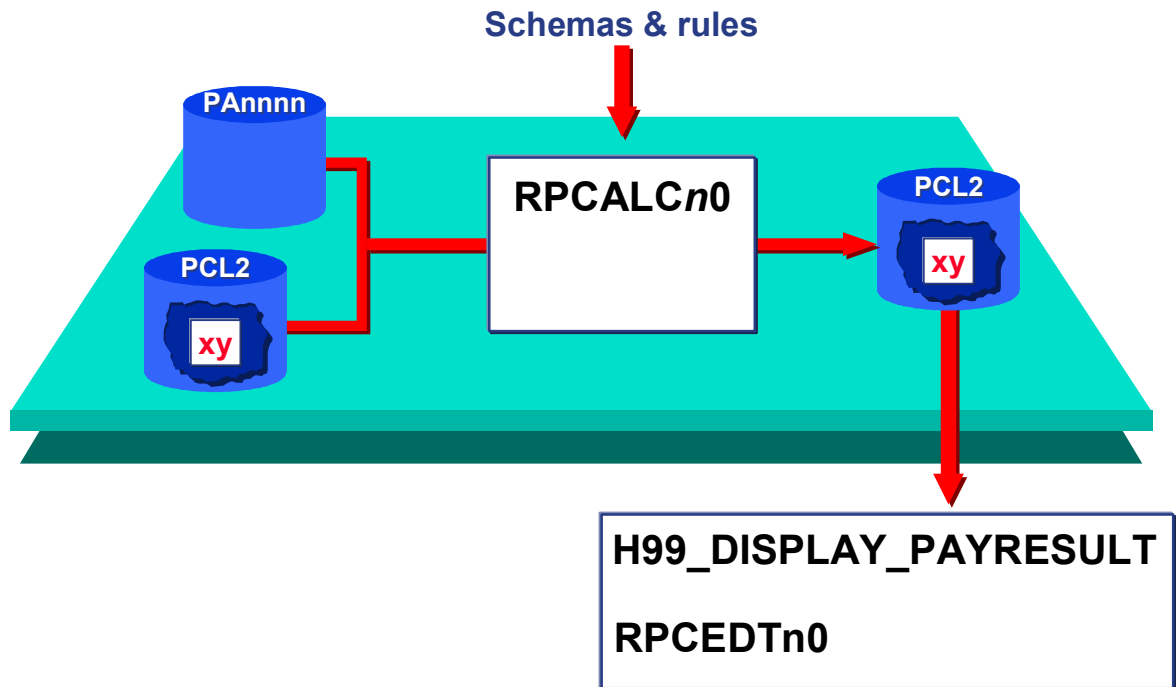
Red:      No results exist

Gray:     No authorization for displaying personnel numbers

- Payroll results are stored as structures and internal tables on the database.

- Each payroll result has a status indicator:

- A    =    Current result
- P    =    Previous result
- O    =    All other results



© SAP AG 1999

- The payroll driver, `RPCALCn0`, uses HR data (stored in the database tables `PAnnnn`) and the last payroll result (stored in the database table `PCL2`) to run the payroll for the specified period
- The program (payroll driver) imports the processing logic in the form of a schema. The schema contains functions that call the subroutines contained in the payroll driver. In many cases, the function is enhanced by rules for specific control of the subroutines.
- The payroll result generated by the payroll driver is stored in cluster `xy` of the database table `PCL2`.
- Report `H99_DISPLAY_PAYRESULT` displays the payroll results for `PCL2` and, from Release 4.6C, replaces reports `RPCLSTxy` and `HxyCLSTR`. The report is used in all country versions and the overview of payroll results is automatically displayed according to the particular country.
- Report `RPCEDTn0`, for example, lists the formatted result as a payroll form ( $n$  = HR country indicator from table `T500L`).

```

*Table containing directory of payroll results
DATA: BEGIN OF RGDIR OCCURS 100.
      INCLUDE STRUCTURE PC261.
DATA: END OF RGDIR.
DATA: COUNTRY LIKE T001P-MOLGA.
...
CALL FUNCTION 'CU_READ_RGDIR'
  EXPORTING
    PERSNR          = PERNR-PERNR
  IMPORTING
    MOLGA           = country
  TABLES
    IN_RGDIR        = RGDIR
  EXCEPTIONS
    NO_RECORD_FOUND = 1
    OTHERS          = 2.
    
```

© SAP AG 1999

- Table RGDIR contains the directory (cluster directory) for all of an employee's payroll results and is contained in cluster CU. A directory entry with the payroll area, for-period, in-period, status indicator, and the five-digit sequence number is required, together with the personnel number, to construct the key for each payroll result for an employee.
- The function module CU\_READ\_RGDIR reads table RGDIR from cluster CU. The personnel number whose payroll directory is to be read is transferred to the function module.
- If the MOLGA parameter is active, the function module returns the HR country indicator.

```

DATA: number LIKE PC261-SEQNR.
. . .
GET PERNR.
. . .
CALL FUNCTION 'CU_READ_RGDIR'
. . .

CALL FUNCTION 'CD_READ_LAST'
  EXPORTING
    BEGIN_DATE      = PN-BEGDA
    END_DATE        = PN-ENDDA
  IMPORTING
    OUT_SEQNR       = number
  TABLES
    RGDIR           = RGDIR
  EXCEPTIONS
    NO_RECORD_FOUND = 1
    OTHERS          = 2.
    
```

© SAP AG 1999

- The function module CD\_READ\_LAST determines the current payroll result for a for-period to be evaluated. To determine the correct start date and end date of the for-period, you specify the period by entering the payroll period in the selection screen. If you specify report class XXM00004 in the attributes of your report, the payroll period is entered and the start date (PN-BEGDA) and the end date (PN-ENDDA) are determined using the Payroll Periods table (T549Q).
- You enter the start and end date of the for-period for the evaluation as well as table RGDIR. The function module then gives you the sequential number (OUT\_SEQNR) for the current (A) result of the for-period.
- You can also use the following function modules:  
 CD\_READ\_PREVIOUS (reads the record that precedes the payroll record)  
 CD\_READ\_PREVIOUS\_ORIGINAL (reads the last original result that precedes the original payroll result)

```

DATA: payroll TYPE PAY99_RESULT.
DATA: rt_line TYPE LINE OF HRPAY99_RT.
. . .
CALL FUNCTION 'PYXX_READ_PAYROLL_RESULT'
  EXPORTING
    CLUSTERID                = 'RX'
    EMPLOYEENUMBER           = p0001-pernr
    SEQUENCENUMBER           = number
*   READ_ONLY_BUFFER         = ' '
*   READ_ONLY_INTERNATIONAL = ' '
*   CHECK_READ_AUTHORITY    = 'X'
    . . .
  CHANGING
    PAYROLL_RESULT           = payroll
  EXCEPTIONS
    . . .
LOOP AT payroll-INTER-RT INTO rt_line.
  WRITE: / rt_line-LGART, ...
ENDLOOP.

```

© SAP AG 2003

- You can use the function module PYXX\_READ\_PAYROLL\_RESULT to read a complete payroll result from the database table PCL2 or from the buffer. The payroll result is then transferred to the parameter PAYROLL\_RESULT. This must be declared in the calling report as a complex structure that corresponds to the structure PAY $ic$ \_RESULT ( $ic$  = ISO code).
- You can use the READ ONLY INTERNATIONAL parameter to specify that only the international part is imported. The READ\_ONLY\_BUFFER parameter prevents the database from being accessed.
- If the parameter CHECK\_READ\_AUTHORITY is activated and set to blank, the cluster authorization check is deactivated. Anonymous evaluations can then be carried out by users without cluster authorizations.

## Structure: PAY99\_RESULT

### EVP

Structure PC261

### INTER Structure PAY99 INTERNATIONAL

#### VERSC

Type PC202

#### RT

Type HRPAY99\_RT

#### CRT

Type HRPAY99\_CRT

#### BT

### NAT

Dummy  
for national  
part

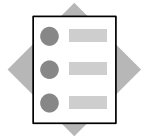
```
DATA: payroll TYPE PAY99_RESULT.
DATA: rt_line TYPE LINE OF HRPAY99_RT.
. . .

* Access to accounting
result data
  WRITE: payroll-INTER-VERSC-FPPER,
  . . .
  LOOP AT payroll-INTER-RT INTO rt_line.
    WRITE: / rt_line-LGART, . . .
  ENDLOOP.
```

© SAP AG 2003

- The data structures for the international payroll results RX are described in the Dictionary in the structure PAY99\_RESULT. The structure contains the components **EVP** (directory information), **INTER** (international), and **NAT** (country-specific part). The INTER and NAT components also contain substructures such as the tables (for example, RT, and CRT) and structures (for example, VERSC) for payroll results. In cluster RX, NAT consists of a dummy field.
- The structures PAY $ic$ \_RESULT ( $ic$  = ISO code, for example, PAYUS\_RESULT for the USA) exist for the country-specific results. Here, the component NAT contains the substructures for the country-specific results.
- If you want to evaluate payroll results, you need a data structure with the type PAY $ic$ \_RESULT. For each table in the payroll results to be processed, you need a header with the type HRPAY $ic\_table\ name$  (for example, HRPAY99\_RT for the results table RT).





**At the conclusion of this topic, you will be able to:**

- **Process payroll results using the logical database for payroll results**



**Payroll results with the table WPBP  
Directory in cluster CU with table RGDIR**



**Directory information from cluster CU  
Table RGDIR**



**Data for work center and basic pay  
Table WPBP from cluster Rx**

© SAP AG 2003

- Payroll results are saved in the cluster table PCL2 to improve performance and free up memory space. However, the logical database requires some of these payroll data in a transparent form, to optimize the performance of data selection. As of the first run in Release 4.6, the payroll driver saves the data in the tables HRPY\_RGDIR and HRPY\_WPBP although this is redundant.
- However, this does not apply for payroll records created in previous releases. To make this data available for evaluation, you must start report H99U\_RGDIR\_WPBP, to fill both tables.
- The evaluation of payroll results is now supported by a logical database that is integrated in the logical database PNP for HR master data.
- The logical database allows you to select and evaluate payroll results based on a wide range of criteria. All information from a payroll record is provided in the components of a complex structure when the **GET PAYROLL** event occurs.
- To assign a report to the logical database for payroll results, enter the selection variant '900' *Selection screen for payroll results* in the report attributes. SAP delivers the sample report *EXAMPLE\_PNP\_GET\_PAYROLL*.

```
TABLES:PERNR, PYORGSCREEN, PYTIMESCREEN.
NODES:PAYROLL TYPE PAY99_RESULT.
INFOTYPES:0001.

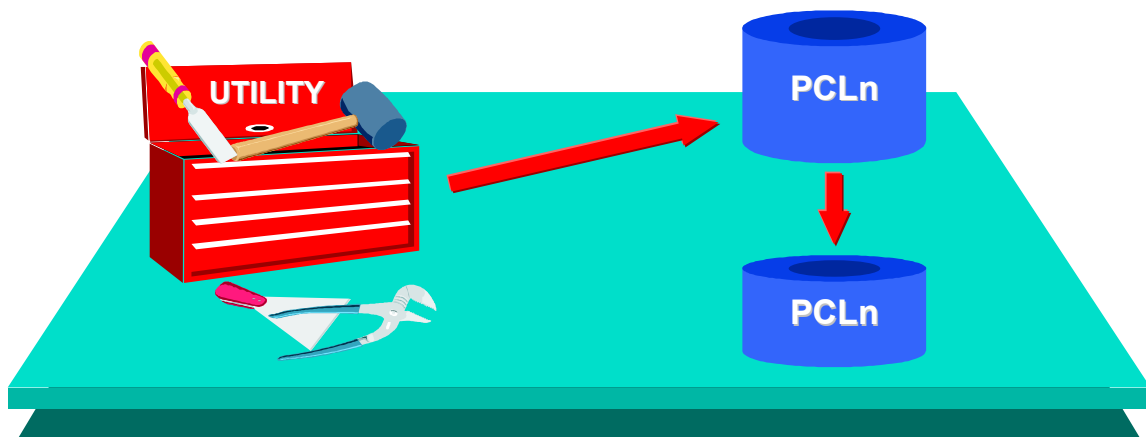
DATA: rt_line TYPE LINE OF HRPAY99_RT.
...
GET PERNR.
    RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGPS PN-ENDPS.
...

GET PAYROLL.
    LOOP AT PAYROLL-INTER-RT INTO rt_line.
        WRITE: / rt_line-LGART, . . .
    ENDLOOP.

GET PERNR LATE.
...
```

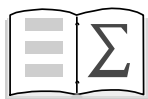
© SAP AG 2003

- PERNR, PYORGSCREEN, and PYTIMESCREEN are declared for the selection screen in the declaration for TABLES.
- NODES is used to declare the country-specific structure PAY $ic$ \_RESULT ( $ic$  = ISO code) of PAYROLL.
- The PAYROLL complex structure is filled when the **GET PAYROLL** event occurs. Note that the GET PAYROLL event only occurs for personnel numbers with a payroll result in the corresponding cluster xx. The result wage types are made available in the PAYROLL-INTER-RT table.
- The evaluation period appears in the fields PN-BEGPS and PN-ENDPS.
- The GET PERNR LATE event is triggered when all selected payroll results are processed. You can issue summaries, for example, when this event occurs.
- You can use the *Result Status* to limit the selection of payroll results.



© SAP AG 2003

- The utility transaction PU00 allows you to delete payroll results in xy clusters for specific personnel numbers.
- You can use the utility programs RPUPnD00 and RPUPnD10 to delete one or more records from any cluster. n stands for database table PCL1 or PCL2.
- You can delete payroll results using the program RPUDEL20.
- **Caution:**  
The deletion utilities should only be used for testing. Archive data to reduce the amount of data in the database.



**You are now able to:**

- **Import cluster data structures**
- **Process payroll results using function modules**
- **Use the logical database for payroll results**





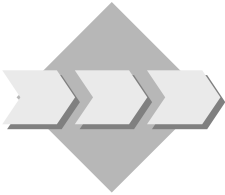
## Unit: Cluster DB Tables in HR

### Topic: Importing Cluster TX



At the end of this exercise, you will be able to:

- Import and display the texts stored in cluster TX for infotype 0040



The departments at your company require a report that provides an overview of the resources loaned to employees. This information is stored in the *Objects on Loan* infotype.

- 1-1 Write a report that only lists employees who have received objects on loan. The report will also print explanations entered by the administrator in the text fields.
  - 1-1-1 The list should contain the following information for each employee:
    - Personnel number
    - Name
    - Start date
    - End date
    - Number
    - Type of object on loan
    - Description of object on loan
  - 1-1-2 Take the name of the employee from the **PERNR-ENAME** field.  
The object on loan corresponds to the subtype for infotype 0040. To determine the subtype text (description of object on loan), read table **T591S** with the keys infotype and subtype. Only display the first 20 characters of the employee name and subtype text.
  - 1-1-3 The include program **RPC1TX00** contains the key for cluster TX, which consists of the fields for the structure **PSKEY**. The data object for the cluster is the table **PTEXT** with the lines LINE.  
Process table P0040 with **LOOP...ENDLOOP** so that the unchanged start and end dates for the infotype records are in the header. If the field **P0040-ITXEX** is not blank, an explanatory text, which is stored in the cluster TX, is entered for the infotype record. If this is the case, enter the key and import cluster TX. If the return code for the import is not 0, display an error message.



## Unit: Exporting/Importing DB Tables in HR

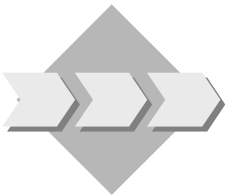
### Topic: Displaying the Last Payroll Result



At the end of this exercise, you will be able to:

- Read the cluster directory with `CU_READ_RGDIR`
- Process the cluster directory with `CD_READ_LAST`
- Read the payroll result with `PYXX_READ_PAYROLL_RESULT`
- Display the wage types from the `RT` results table

Your company's payroll department needs a list of wage types from the results table for the current (last) payroll result in a for-period.



- 1-1 Write a report to read and display the results of international payroll to cluster RX. The report should evaluate the last payroll result for the for-period. Test your program using the personnel number assigned by your trainer.

The created list should contain the following information for each employee:

Line 1:   - Personnel number  
          - Name  
          - Personnel area

Line 2:   - For-period

Line 3:   - In-period

Line 4:   - Wage type  
          - Wage type text  
          - Amount

- 1-1-1 Take the name of the employee from the **PERNR-ENAME** field.  
Information on the for-period and in-period can be found in the fields for the **VERSC** structure, which is a structure in cluster RX.

- 1-1-2 First use the function module **CU\_READ\_RGDIR** to read table `RGDIR` from cluster CU. To determine the sequential number for the last payroll result, transfer table `RGDIR` to the function module **CD\_READ\_LAST**.

- 1-1-3 Then read the payroll result for the sequential number you have determined using the function module **PYXX\_READ\_PAYROLL\_RESULT**.

- 1-1-4 The wage types and amounts are saved in the **RT** table. The texts belonging to the wage types can be found in table **T512T**.

Only display the first 15 characters of the employee name.





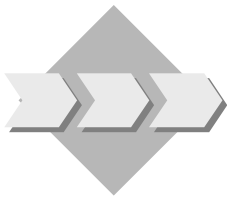
## **Unit: Exporting/Importing DB Tables in HR**

### **Topic: Logical Database for Payroll Results**



At the end of this exercise, you will be able to:

- Process payroll results using the logical database for payroll results



Your company's payroll department needs a list of wage types from the results table for payroll accounting.

- 1-1 Copy the program from the previous exercise and modify it so that you now use the logical database for payroll results.





## Unit: Cluster DB Tables in HR

### Topic: Importing Cluster TX

```
REPORT zpsol060.
```

```
TABLES: pernr,  
        pcl1,  
        pcl2,  
        t591s.                                "Subtype Table  
  
INFOTYPES: 0001,                             "Org.Assignment  
           0040.                             "Objects on Loan  
  
INCLUDE: rpc1tx00.                            "Definition Cluster TX  
INCLUDE: rpppxd00.                            "Buffer-Definition (I)  
DATA: BEGIN OF COMMON PART buffer.  
INCLUDE: rpppxd10.                            "Buffer-Definition (II)  
DATA: END   OF COMMON PART buffer.  
  
GET pernr.  
  rp_provide_from_last p0001 space pn-begda pn-endda.  
  LOOP AT p0040 WHERE endda GE pn-begda  
        AND   begda LE pn-endda.  
    PERFORM read_t591s USING p0040-infty p0040-subty.  
    PERFORM print_data.  
    IF NOT p0040-itxex IS INITIAL.  
      PERFORM read_cluster.  
      PERFORM print_cluster.  
    ENDIF.  
  ENDLOOP.  
  
END-OF-SELECTION.
```

```

*-----*
*          FORM READ_T591S
*-----*
*          Read Subtype Texts
*-----*
FORM read_t591s USING info_type sub_type.
  CHECK t591s-sprsl NE sy-langu
    OR t591s-infty NE info_type
    OR t591s-subty NE sub_type.
  SELECT SINGLE * FROM t591s
    WHERE sprsl      = sy-langu
    AND   infty      = info_type
    AND   subty      = sub_type.
ENDFORM.

```

```

*-----*
*          FORM PRINT_DATA
*-----*
FORM print_data.
  FORMAT INTENSIFIED ON.
  WRITE: / pernr-pernr,
    (20) p0001-ename,
    p0040-begda,
    p0040-enddda,
    p0040-anzkl,
    p0040-leihg,
    (20) t591s-stext.
  FORMAT INTENSIFIED OFF.
ENDFORM.

```

```

*-----*
*          FORM READ_CLUSTER
*-----*
*          Read Cluster TX
*-----*
FORM read_cluster.
  CLEAR tx-key.
  MOVE-CORRESPONDING p0040 TO tx-key.
  rp-imp-cl-tx.
  IF sy-subrc NE 0.
    WRITE: / 'Cluster TX was deleted.'(001).
  ENDIF.
ENDFORM.

```

```

*-----*
*          FORM PRINT_CLUSTER
*-----*
FORM print_cluster.
  SKIP.
  LOOP AT ptext.
    WRITE: / ptext-line.
  ENDLOOP.
  SKIP.
ENDFORM.

*-- Subroutines
INCLUDE: rpppxm00.                "Buffer subroutines

```



## Unit: Exporting/Importing DB Tables in HR

### Topic: Displaying the Last Payroll Result

```
REPORT zpsol070.
TABLES: pernr,
        t512t.                "Wage type texts
INFOTYPES: 0001.              "Organizational Assignment

*Table data containing directory to PCL2 payroll results file
DATA: BEGIN OF rgdir OCCURS 100.
      INCLUDE STRUCTURE pc261.
DATA: END OF rgdir.

DATA: payroll TYPE pay99_result.
DATA: rt_header TYPE LINE OF hrpay99_rt.

DATA: country LIKE t001p-molga,
      number  LIKE pc261-seqnr. "Number of last payroll result

GET pernr.
  rp_provide_from_last p0001 space pn-begda pn-endda.

  CALL FUNCTION 'CU_READ_RGDIR'
    EXPORTING
      persnr      = p0001-pernr
    IMPORTING
      molga       = country
    TABLES
      in_rgdir    = rgdir
    EXCEPTIONS
      no_record_found = 1
      OTHERS         = 2.

  IF sy-subrc = 1.
    WRITE: / 'No records found for '(001), pernr-pernr.
    REJECT.
  ENDIF.
```

```

CALL FUNCTION 'CD_READ_LAST'
  EXPORTING
    begin_date      = pn-begda
    end_date        = pn-endda
  IMPORTING
    out_seqnr       = number
  TABLES
    rgdir           = rgdir
  EXCEPTIONS
    no_record_found = 1
    OTHERS          = 2.

IF sy-subrc = 1.
  WRITE: / 'No payroll result found for'(002), pn-paper.
ELSE.
  CALL FUNCTION 'PYXX_READ_PAYROLL_RESULT'
    EXPORTING
      clusterid      = 'RX'
      employeenumbr  = p0001-pernr
      sequencenumbr   = number
*    READ_ONLY_BUFFER      = ' '
*    READ_ONLY_INTERNATIONAL = ' '
*    CHECK_READ_AUTHORITY  = 'X'
*  IMPORTING
*    VERSION_NUMBER_PAYVN   =
*    VERSION_NUMBER_PCL2    =
  CHANGING
    payroll_result      = payroll
  EXCEPTIONS
    illegal_isocode_or_clusterid = 1
    error_generating_import      = 2
    import_mismatch_error        = 3
    subpool_dir_full             = 4
    no_read_authority            = 5
    no_record_found              = 6
    versions_do_not_match        = 7
    OTHERS                      = 8.

IF sy-subrc = 0.
  PERFORM print_rx.
ELSE.
  WRITE: / 'Result could not be read (003).
ENDIF.
ENDIF.

```

```

*-----*
*          FORM PRINT_RX                      *
*-----*
*          Print Payroll Result                *
*-----*

```

FORM print\_rx.

FORMAT INTENSIFIED ON.

WRITE: / p0001-pernr,  
     p0001-ename(15),  
     p0001-werks,  
     p0001-btrtl.

FORMAT INTENSIFIED OFF.

SKIP 1.

WRITE: / 'For period/payroll area: '(004),  
     30 payroll-inter-versc-fpper+4(2),  
     payroll-inter-versc-fpper+0(4),  
     payroll-inter-versc-abkrs,  
     / 'In-period/payroll area: '(005),  
     30 payroll-inter-versc-inper+4(2),  
     payroll-inter-versc-inper+0(4),  
     payroll-inter-versc-iabkrs.

SKIP 1.

WRITE: 'Results table: '(006).

SKIP 1.

LOOP AT payroll-inter-rt INTO rt\_line.

PERFORM re512t USING payroll-inter-versc-molga  
                     rt\_line-lgart.

WRITE: / rt\_line-lgart,  
     t512t-lgtxt,  
     rt\_line-betrp CURRENCY rt\_line-amt\_curr.

ENDLOOP.

ENDFORM.

```

*-----*
*          FORM RE512T                      *
*-----*
*          Read Wagetyp Texts                *
*-----*

```

FORM re512t USING value(country\_grouping)  
                     value(wtype).

CHECK t512t-sprsl NE sy-langu  
     OR t512t-molga NE country\_grouping  
     OR t512t-lgart NE wtype.

SELECT SINGLE \* FROM t512t  
             WHERE sprsl EQ sy-langu  
             AND molga EQ country\_grouping  
             AND lgart EQ wtype.

IF sy-subrc NE 0.

CLEAR t512t.

ENDIF.

ENDFORM.





## Unit: Exporting/Importing DB Tables in HR

### Topic: Logical Database for Payroll Results

REPORT zpsol080.

TABLES: pernr, pyorgscreen, pytimescreen, t512t.

NODES: payroll TYPE pay99\_result.

INFOTYPES: 0001. "Organizational Assignment

DATA: rt\_header TYPE LINE OF hrpay99\_rt.

GET pernr.

rp\_provide\_from\_last p0001 space pn-begps pn-endps.

WRITE: / p0001-pernr,  
p0001-ename(15),  
p0001-werks,  
p0001-btrtl.

SKIP 1.

GET payroll.

WRITE: / 'For-period:'(001),  
15 payroll-inter-versc-fpper+4(2),  
payroll-inter-versc-fpper+0(4),  
/ 'In-Period: '(002),  
15 payroll-inter-versc-inper+4(2),  
payroll-inter-versc-inper+0(4).

SKIP 1.

WRITE: 'Results table: '(003).

SKIP 1.

LOOP AT payroll-inter-rt INTO rt\_line.

PERFORM re512t USING payroll-inter-versc-molga  
rt\_line-lgart.

WRITE: / rt\_line-lgart,  
t512t-lgtxt,  
rt\_line-betrg CURRENCY rt\_line-amt\_curr.

ENDLOOP.

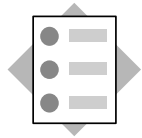
```

*-----*
*          FORM RE512T
*-----*
*          Read Wage Type Texts
*-----*
FORM re512t USING value(country_grouping)
                value(wtype) .
CHECK t512t-sprsl NE sy-langu
      OR t512t-molga NE country_grouping
      OR t512t-lgart NE wtype.
SELECT SINGLE * FROM t512t
                WHERE sprsl EQ sy-langu
                AND   molga EQ country_grouping
                AND   lgart EQ wtype.
IF sy-subrc NE 0.
  CLEAR t512t.
ENDIF.
ENDFORM.

```

## **Contents:**

- **Data model**
- **Technical structures of infotypes in Personnel Planning**



**At the conclusion of this unit, you will be able to:**

- **Explain the data model of Personnel Planning**
- **Describe the data structures of Personnel Planning**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration

 Data Structures in Personnel Planning

The Logical Databases PNP/PNPCE

The Logical Database PCH

Join and Projection

Customer-Specific Infotypes

Specific Commands

HR ABAP Features

Repetitive Structures and List Display



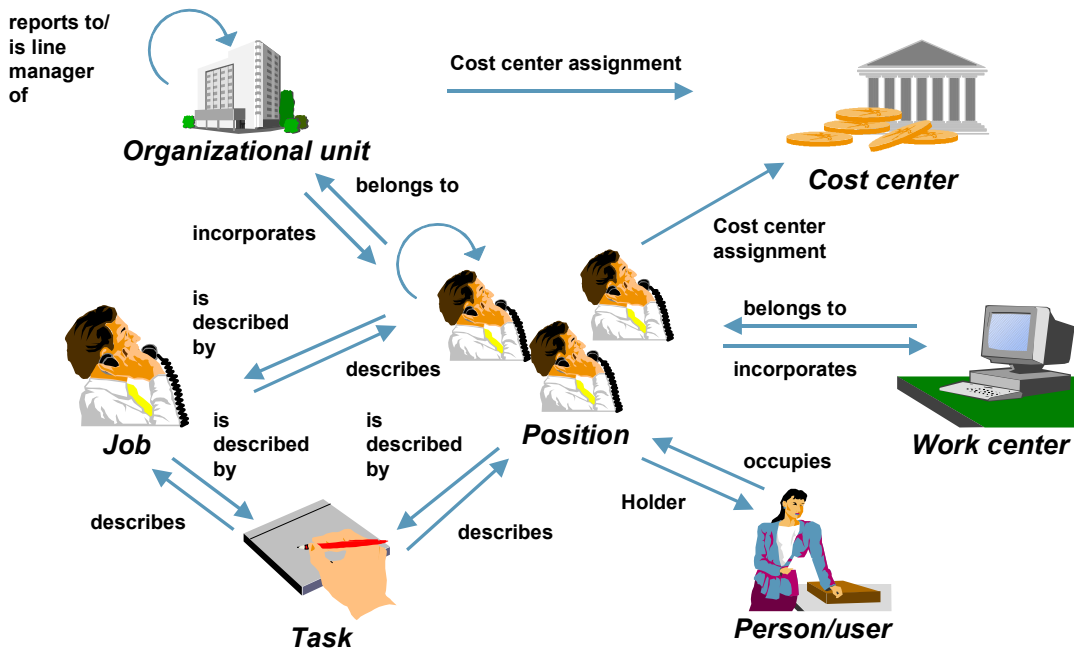
- You want to draw up your own evaluations of **Personnel Administration data**. To do this, you must learn about the data model and data structures in **Personnel Planning**.



**At the conclusion of this topic, you will be able to:**

- **Explain the Personnel Planning data model**

## ● Object-oriented design - *object types*

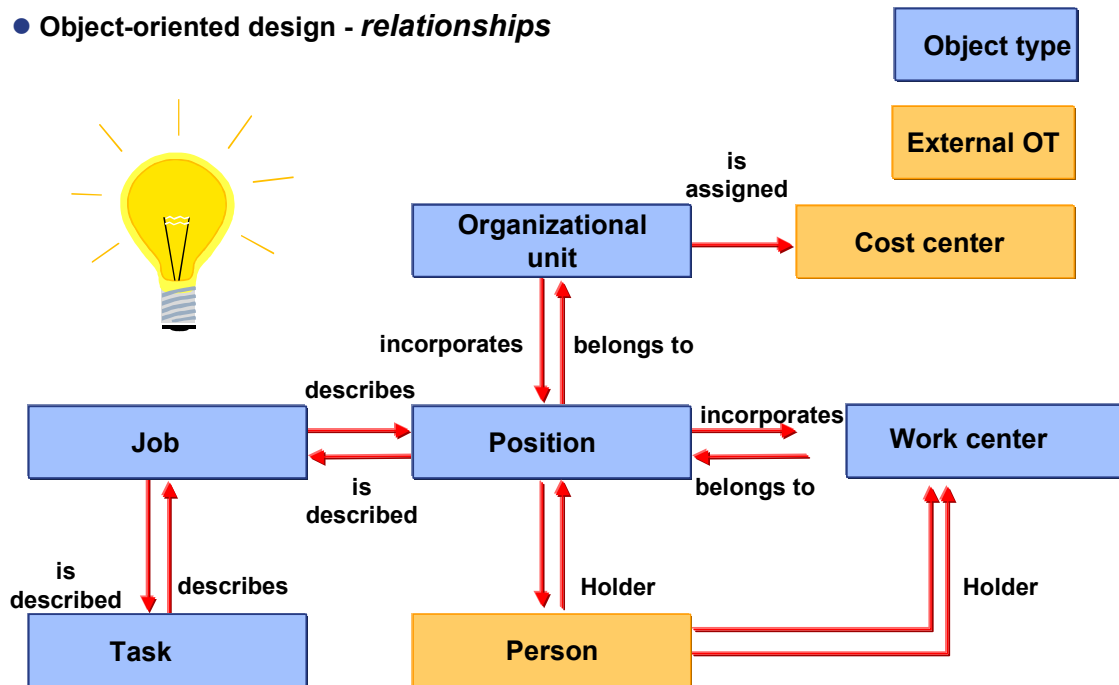


© SAP AG 2003

- The Personnel Planning data model is based on an **object-oriented design**.
- This design is based on the following concepts:
  - Object types
  - Relationships
  - Infotypes
- In Organizational Management, each element in an organization is displayed as an independent object with individual characteristics. These objects are created and maintained separately. Relationships are used to link one object to the other (see graphic). This creates a network that is flexible enough to facilitate personnel planning, projections, and evaluations.
- The cost center is an external object type because it is not maintained in Organizational Management.
- You can use Customizing to enhance the existing data model by defining new object types, for example, and establishing new relationships between the various object types. Each standard object type consists of two letters, and the customer namespace is 00 to 99.
- This data model (object types and relationships) also constitutes the basis of other applications within Personnel Planning, such as Training and Event Management (business event hierarchies) and Personnel Development (for example, qualification catalog).



- Object-oriented design - *relationships*



© SAP AG 2003

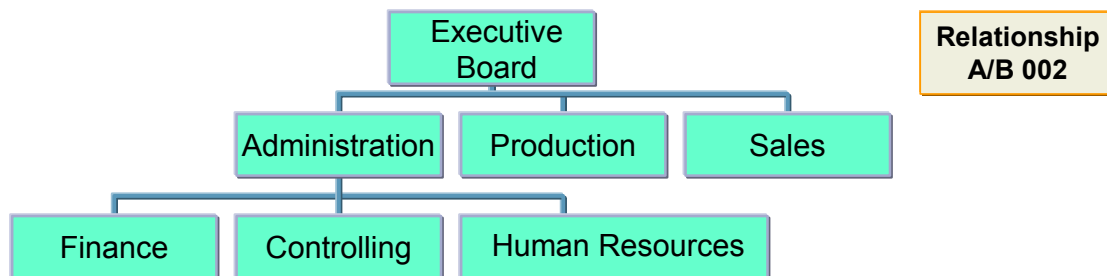
- You can use relationships to define links between individual object types in the system.
- Relationships are usually defined in two directions: *A = bottom up* and *B = top down*.
- For example: Relationship "003" ("incorporates/belongs to") is defined in the following directions:
  - A003 "belongs to"* and *B003 "incorporates"*
- However, this convention is an optional suggestion that you may choose to adopt when you first create a relationship. It is crucial that you do not change or invert a relationship to directions A and B at a later stage.
- For internal object types, each relationship is also stored physically on the database in both directions. For relationships to external object types, only one direction may be stored on the database, depending on the key structure of the external object type. In this case, only the relationship to the external object is stored (**external object types without the inverse relationship**).

**An organizational unit "reports to" another organizational unit**

For example: Finance "reports to" Administration

**An organizational unit "is the line supervisor of" other organizational units**

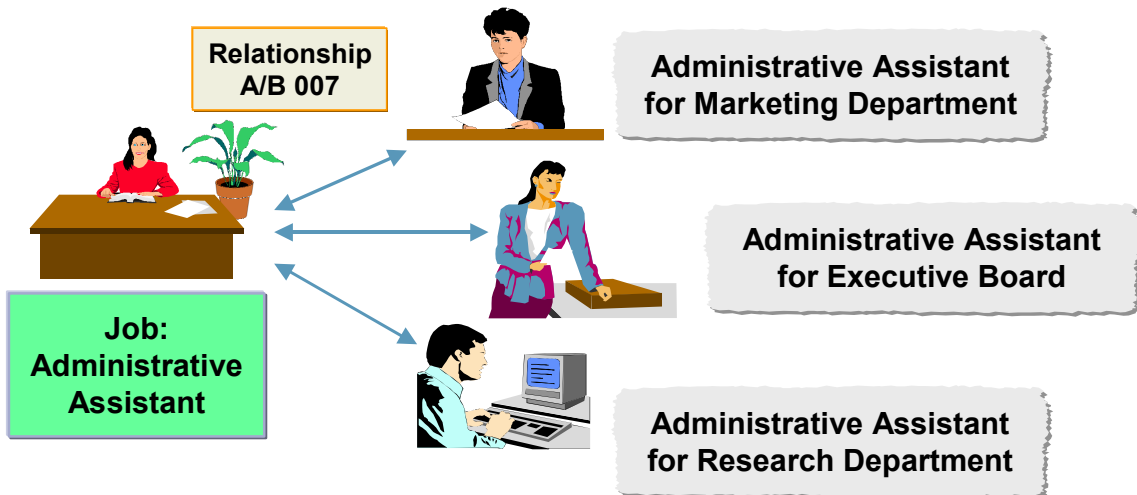
For example: Executive Board "is the line supervisor of" Administration, Production, and Sales



© SAP AG 2003

- Relationships are usually stored on the database in two directions.

**The job "describes" the position.  
The position "is described by" the job.**

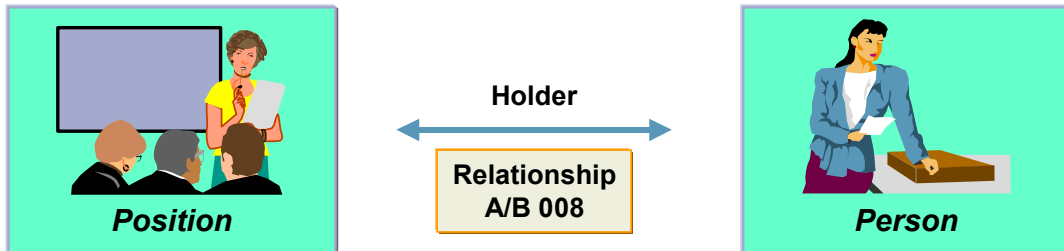


© SAP AG 2003

- The term "job" is used as a general classification, such as "secretary" or "head of department".
- More than one employee can have the same job. For example, twenty employees can have the job of secretary.
- A position refers to the individual assignment of an employee in the company, for example, sales manager, secretary of the marketing departments. You create positions and establish relationships between them to map the line structure (organization chart) in your company.

### The position is assigned to a person

For example: Ms Smith is the "holder" of the position of HR manager.



**The relationship between position and person creates integration in master data.**

© SAP AG 2003

- If Organizational Management is actively integrated with master data, the holder relationship between position and person is also stored for the person in infotype 0001 *Organizational Assignment*, although this is redundant.

**Object characteristics are maintained in the form of infotypes.**

## Main characteristics

- **Object** (1000)
- **Relationship** (1001)

## Other characteristics

- **Description** (1002)
- **Working time** (1011)
- **Cost planning** (1015)
- **Address** (1028)

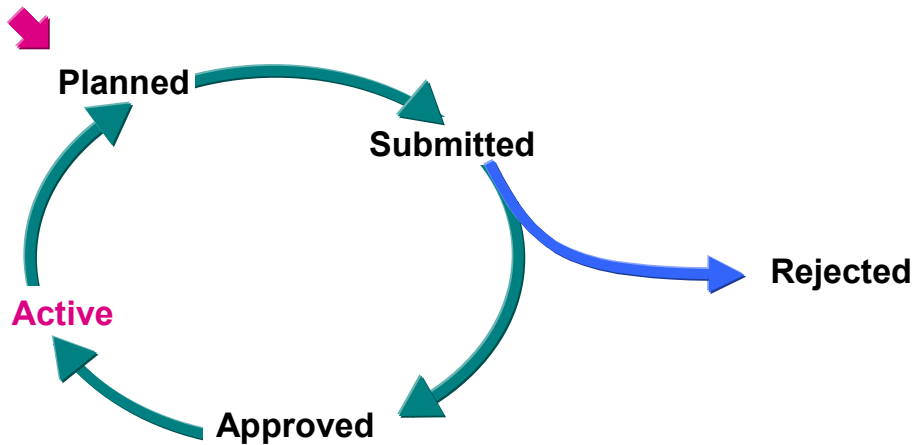
## Examples

- **Organizational unit, position, job**
- **Belongs to, reports to**
- **General text**
- **Planned working hours**
- **Cost elements**
- **Main address, second address**

© SAP AG 2003

- The *Object* infotype performs key functions. It enables you to create new objects, such as organizational units and jobs, and determines the validity period of all other infotypes that describe the characteristics of existing objects.
- The *Object* infotype (1000) defines an object's existence. You can use it to store a short text and a long text for the object.  
An object is uniquely identified by the following:
  - Plan version
  - Object type
  - Object identification (ID number)
- Once you have used this infotype to create an object, you can use other infotypes such as the *Relationships* (1001) and *Description* (1002) to describe its characteristics.
- Records stored in the *Relationships* infotype (1001) serve as a descriptive and functional link between objects. For example:
  - A person occupies a position.
  - A position belongs to an organizational unit.
  - An organizational unit is assigned to a cost center in CO.

- **Object** (Infotype 1000)
- **Relationship** (Infotype 1001)



© SAP AG 2003

- One of the following statuses is assigned to each object/infotype:
- **Active** means that an object or infotype record can be currently be used.
- **Planned** means that an object or infotype record has been proposed but is not yet active, which means it cannot be used.
- **Submitted** means that an object or infotype record is currently being checked by a person or group of persons before it is approved or rejected.
- **Approved** means that a submitted object/infotype record has been accepted.
- **Rejected** means that a submitted object/infotype record has not been approved.
- You can create objects with the status *Planned* or *Active*.

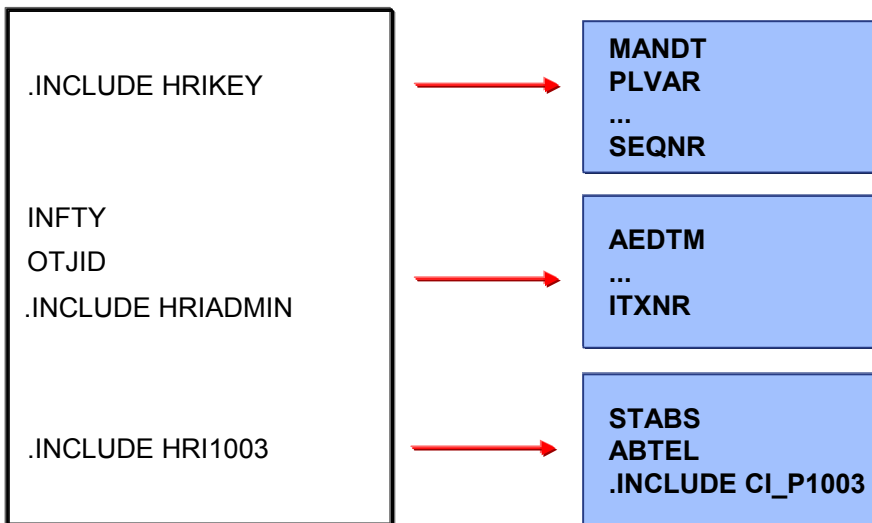


**At the conclusion of this topic, you will be able to:**

- **Describe the data structures of Personnel Planning**

- **Transparent tables *Dictionary field definition***

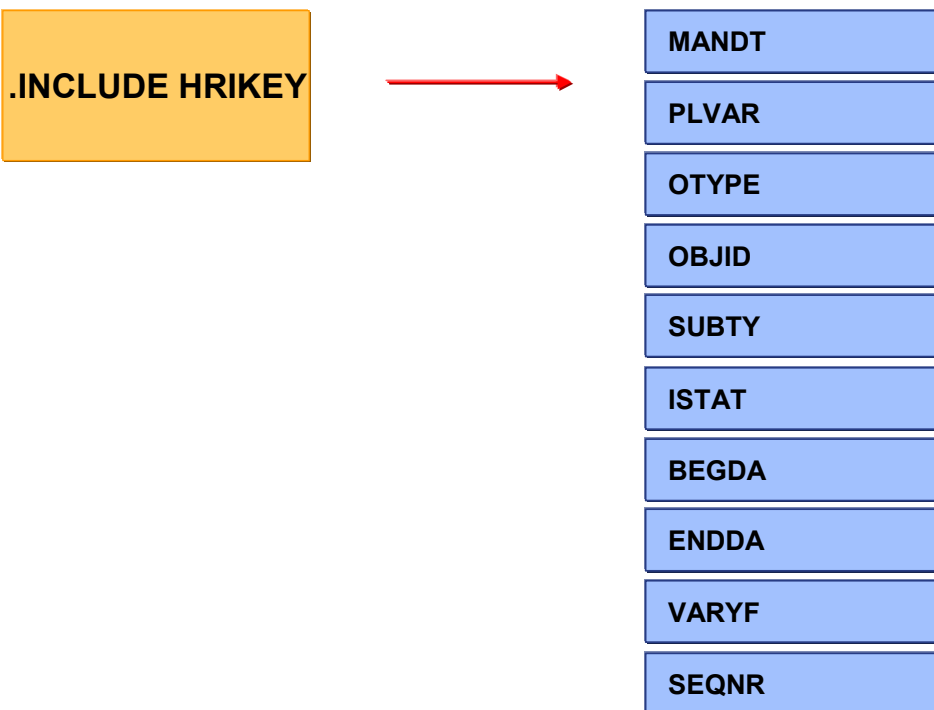
Example: HRP1003



© SAP AG 2003

- With the exception of HRP1000 and HRP1001, each infotype table HRPnnnn is defined in the Dictionary primarily using the following three include structures:
  - **PAKEY:**  
Contains the infotype table key fields
  - **HRIADMIN:**  
Contains administrative information on the infotype record (amongst other things, the date the last change was made and the name of the user who made the change)
  - **HRInnnn:**  
Contains the infotype-specific fields that holds the infotype's actual information and data
- The structure HRInnnn is also included by the structure Pnnnn so that the transparent table HRPnnnn and the logical structure Pnnnn are always consistent with each other.





© SAP AG 2003

- The primary key of the transparent table HRPnnnn is made up of the following key fields:
  - **MANDT**: A participant in the system that is legally and organizationally self-contained
  - **PLVAR**: One- or two-digit alphanumeric **plan version key** that distinguishes different organizational plans
  - **OTYPE**: One- or two-digit alphanumeric **object type key** (for example, 'S' for position, 'Q' for qualification, 'E' for event)
  - **OBJID**: Eight-digit numerical, non-mnemonic key that displays an individual object (for example, an organizational unit, a qualification or an event)
  - **SUBTY**: **Subtypes** are subdivisions of infotypes. The subtypes of an infotype can have different time constraints and create their own history.
  - **ISTAT**: One-digit numeric key that represents a planning status. All objects and their descriptive infotypes have a status. You can use this status to guide objects and infotypes through a planning cycle.
  - **BEGDA**: Valid from date
  - **ENDDA**: Valid to date
  - **VARYF**: Variation field (contains the target object for relationships)
  - **SEQNR**: The **sequential number** allows you to differentiate infotype records that have the same key and the time constraint '3' (any number of valid data records of an infotype at any one time). It is automatically assigned by the system.

**.INCLUDE HRIKEYL**



MANDT

PLVAR

OTYPE

OBJID

SUBTY

ISTAT

BEGDA

ENDDA

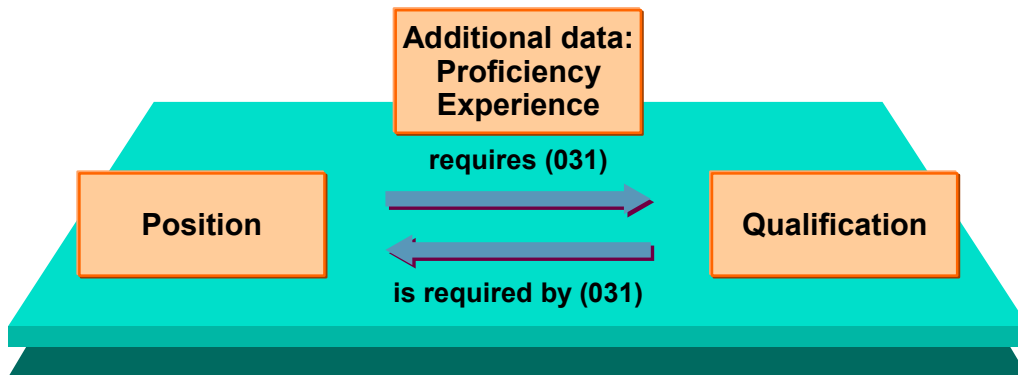
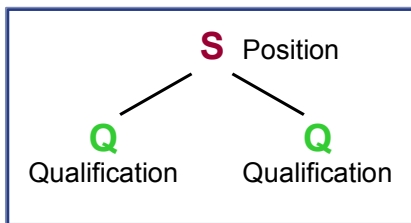
LANGU

DUMMY

SEQNR

© SAP AG 2003

- **Language-specific infotypes** are infotypes that contain one or more text fields (TEXT type) that can be translated.
- A language-specific infotype is labeled as such in the *LANGU* ("Language-specific") field in table T777D.
- Examples of language-specific infotypes in the standard system include:
  - Infotype 1000 (Object): Object short and long texts can be translated
  - Infotype 1002 (Description): A text of any length can be translated
- To store the language of the infotype record in language-specific infotype records, an easily changeable primary key structure (*HRIKEYL*) is used for language-specific infotypes, compared with the non-language-specific infotype (*HRIKEY*). The language of the infotype record is stored in the first position of the VARYF field, that is, instead of the ten-digit VARYF field, the *HRIKEYL* primary key structure contains the following fields:
  - *LANGU* (language ID) with length of 1
  - *FILLER* (filler field) with a length of 9
    - In this way, the key of a language-specific infotype record differs from the key of its translated infotype records in the *LANGU* field. All other key fields are identical.

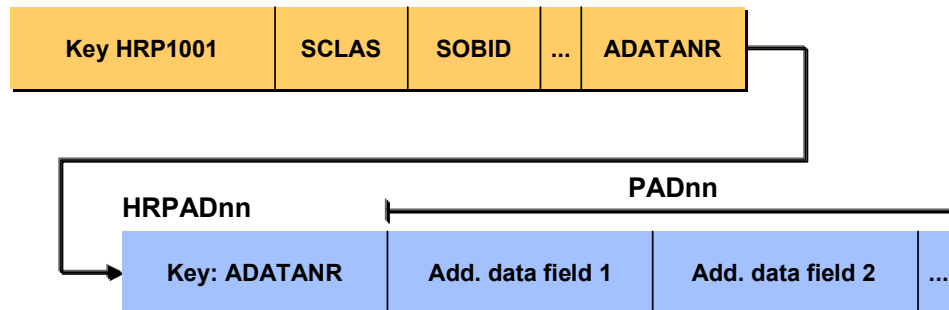


© SAP AG 2003

- Additional data on relationships comprises attributes that cannot be stored in isolation as infotypes for the objects in question because they refer exclusively to the concrete relationship between two objects.
- For example:  
The relationship 031 "requires/is required by" between a position and a qualification. A position may require the *Language skills* qualification, with the qualification providing no information on the level of language skills. The proficiency of language skills is stored as additional information in the relationship between the position and the qualification.

- **Technical implementation - *additional data***

Infotype **HRP1001**



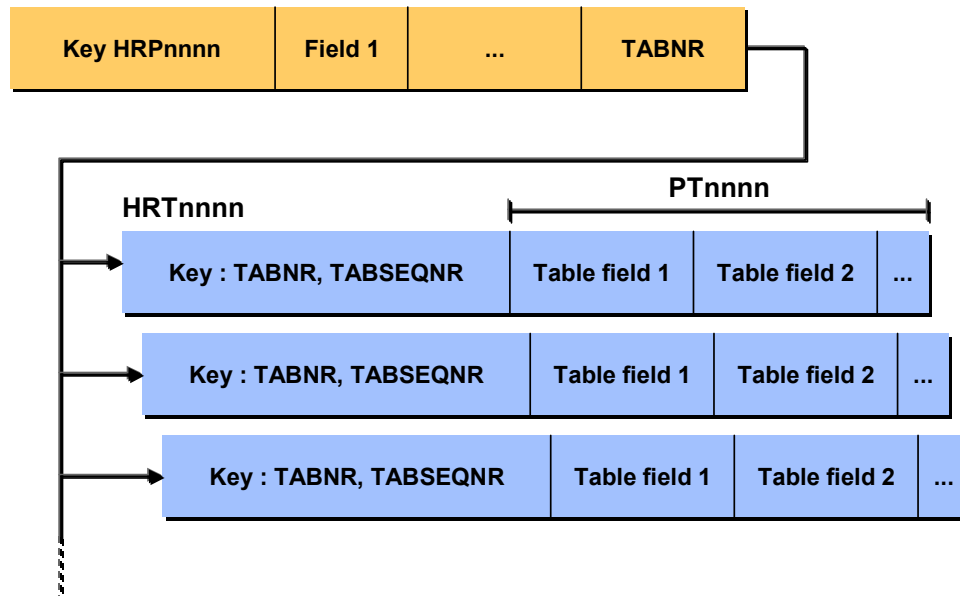
**Table T77AD:** Name of database table for an additional data structure

© SAP AG 2003

- As the structure of additional data depends on the type of relationship, the additional data of the *PADnn* structure must be stored in the separate table *HRPADnn*.
- Table T77AD is used to store the assignment between the database table and the additional data structure.
- The relationship between the relationship record (table *HRP1001*) and the additional data record (table *HRPADnn*) is determined by the **additional data pointer** (field *ADATANR*). The additional data pointer is determined internally by SAP using internal number distribution and cannot be changed externally.
- In the same way as for infotype records, the logical structure *PADnn* should always be used when you work with additional data, to avoid direct database accesses of table *HRPADnn*.
- In the *P1001* logical structure of a *relationship record* (infotype 1001), the additional data is stored in the *ADATA* field. The content of the *P1001-ADATA* field is transferred depending on the type of relationship in the corresponding additional data structure *PADnn* (in accordance with table *T77AR*).

- **Technical implementation - *table infotypes***

Table infotype **HRPnnnn**



© SAP AG 2003

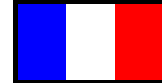
- **Table infotypes** refer to infotypes whose data has a repetitive structure of any length (for example, text with any number of lines, business event schedules with any number of days). To save repetitions of any length, the data part must be stored with the repetitive structure ("table part", described by the *PTnnnn* logical structure) in the separate table *HRTnnnn*. The name of this transparent table is stored in the table *T777D* in the *TBTAB* field ("Database table for table infotype"). In this way, the *T777D-TBTAB* field identifies a table infotype.
- The relationship between the primary record (table *HRPnnnn*) and the table part (table *HRTnnnn*) is determined by a **table pointer** (*TABNR* field). In the same way as the additional data pointer, the table pointer is determined internally by SAP using internal number distribution and cannot be changed externally.
- In the same way as for infotype records, when you are working with table infotypes, the *PTnnnn* logical structure should always be used for the *table part*, to avoid direct database accesses of *table HRTnnnn*.

- **Definition *country-specific infotypes***

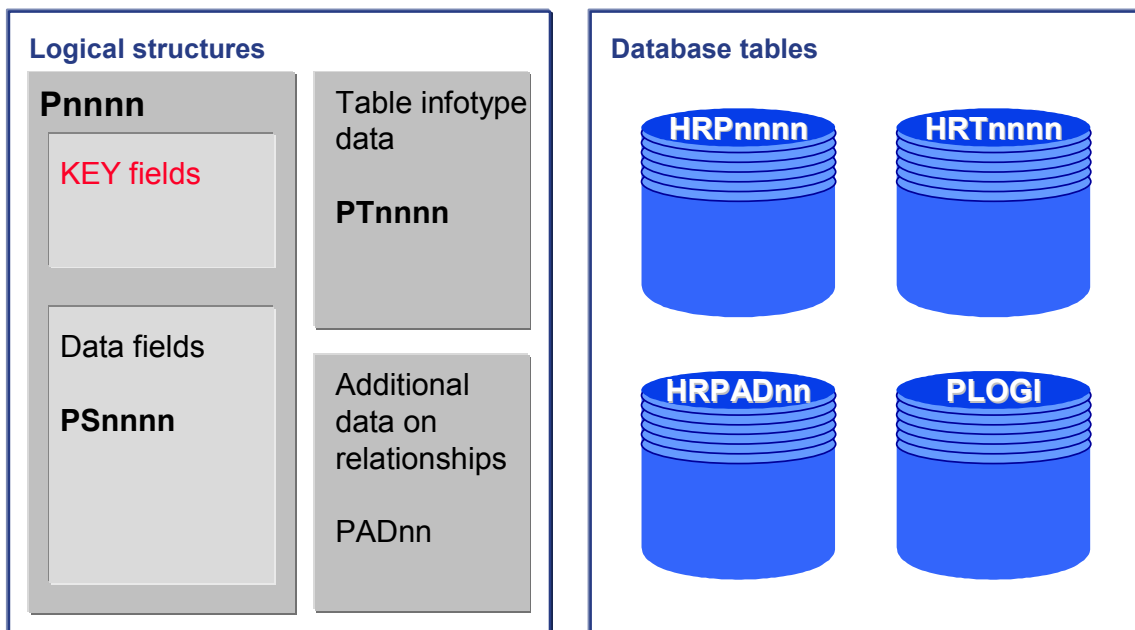
- ➔ **Country-specific infotypes** = Infotypes that are only relevant for a specific country and should therefore only appear when the corresponding country indicator is entered in general infotype overviews

- ➔ **Table T77NI** (IMG path: "Infotype maintenance" - "Maintain country-specific infotypes")

Infotype	Country ID
1600	FR
1601	FR
1610	US
1611	US
...	

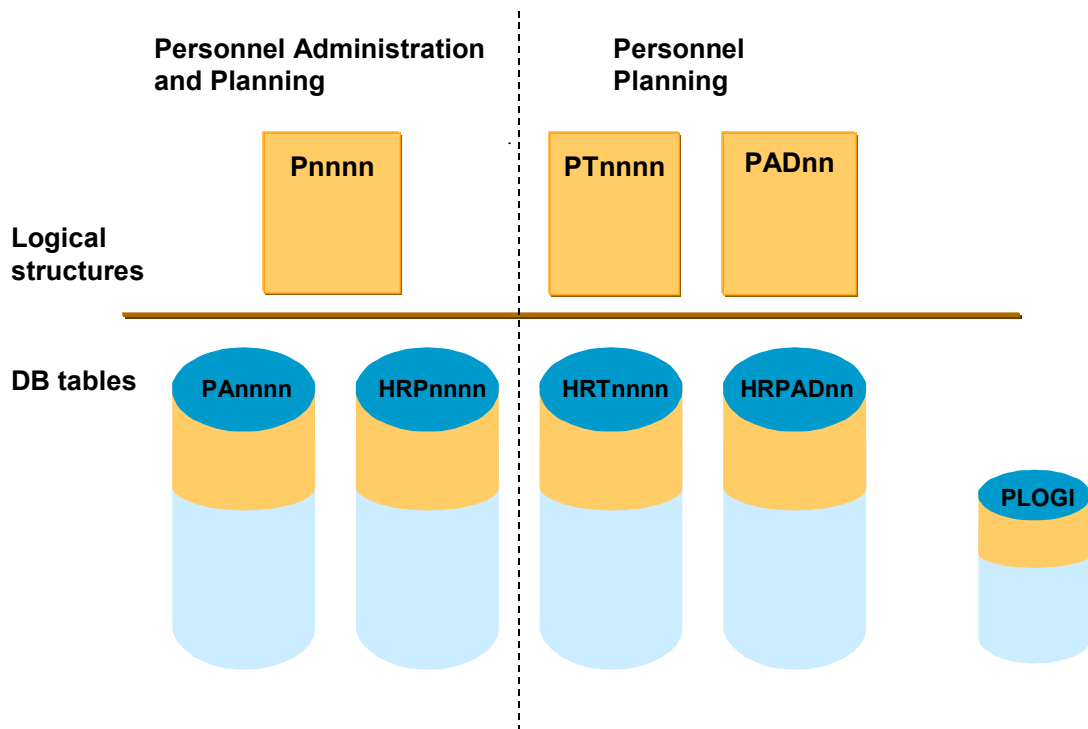


- The *NAT\_INFITY* ("Country-specific infotype") field is used to label a country-specific infotype in table *T777D*.
- The country-specific infotype is assigned to one or more country IDs (from table *T005*) in the table *T77NI*.
- When you enter the country ID in detailed maintenance (using the corresponding dialog box) and in the report *RHDESC00* (using a parameter), the corresponding country-specific infotypes are also displayed in infotype overviews.
- You can use the *PN1 SPA/GPA* parameter to set user-specific default settings for the country ID. You can store up to six country IDs here. You can use the *PNALL SPA/GPA* parameter to activate the general display of all country-specific infotypes.



© SAP AG 2003

- The **Pnnnn** logical structure contains an infotype's key and data fields. During interface definition, it is used in Personnel Administration between programs and between programs and screens.
- The **PTnnnn** structure contains the table infotype data fields for table infotypes.
- The **PADnn** structure defines additional data for specific relationships.
- The transparent **HRPnnnn** tables contain the infotypes used in Personnel Planning. The table parts of table infotypes are stored in the **HRTnnnn** tables. Additional data on relationships is stored in the **HRPADnn** tables.
- The transparent table **PLOGI** contains an index of all objects on the Personnel Planning database. You can use the **RHPLOGI0** report for display/set up. This object directory forms the basis of all evaluations using the **PCH** logical database.



© SAP AG 2003

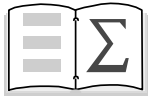
- The following elements exist for each nnnn infotype:
  - A *Pnnnn* logical structure
  - A *PAnnnn* transparent table (administration infotype) or *HRPnnnn* (planning infotype)
- The following elements also exist for each table infotype nnnn in Personnel Planning:- A logical structure *PTnnnn* for the table part- A *HRTnnnn* transparent table for the table part
- Each relationship type with additional data is also assigned the following:
  - A *PADnn* logical structure for additional data of this relationship type
  - A *HRPADnn* transparent table for the additional data of this relationship type
- The logical structures are crucial for programming in HR. Direct accesses to transparent tables are not required and are therefore to be avoided.
- The *PLOGI* transparent table provides a redundant directory of all Personnel Planning objects for the 1000 infotype and is used for evaluations.
- For each Personnel Planning object, the table contains just one entry with the following object ID:
- MANDT (Client)/PLVAR (Plan version)/OTYPE (Object type)/OBJID (Object ID)
- You can use the *RHPLOGI0* report to display and set up the table.



Element	Definition	Included	Changeability
<b>HRIKEY(L)</b>	Structure	-	No
<b>HRInnnn</b>	Structure	CI_Pnnnn	HR10nnn-HR18nnn: Only CI_Pnnnn HR19nnn: Yes
<b>Pnnnn</b>	Structure	HRInnnn	P0nnn-P8nnn: No Only HR19nnn
<b>HRPnnnn</b>	Transparent table	HRIKEY or HRIKEYL	HRP0nnn-HRP8nnn: No Only HR19nnn
<b>PTnnnn</b>	Structure	-	PT0nnn-PT8nnn: No PT9nnn: Yes
<b>HRTnnnn</b>	Transparent table	PTnnnn	HRT0nnn-HRT8nnn: No Only PT9nnn
<b>CI_Pnnnn</b>	Structure	-	Yes

© SAP AG 2003

- The table displayed above provides an overview of the data structures and transparent tables that belong to an infotype in Personnel Administration.



**You are now able to:**

- **Explain the Personnel Planning data model**
- **Describe the data structures of Personnel Planning**

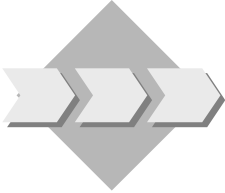


## Unit: Data Structures in Personnel Planning



At the end of this exercise, you will be able to:

- Explain the Personnel Planning data model
- Describe the data structures of Personnel Planning



You want to draw up your own evaluations of Personnel Administration data. To do this, you must learn about the data model and data structures in Personnel Planning.

1-1 The Personnel Planning data model is based on an **object-oriented design**.

1-1-1 Name the concepts that form the basis of this design:

---

---

1-1-2 How can you extend the Personnel Planning data model?

---

---

1-2 What is meant by additional data on relationships?

---

---

1-2-1 In which logical structure is additional data stored?

---

---

1-3 What is meant by a table infotype?

---

---

1-3-1 Which fields does the PTnnnn structure contain?

---

---





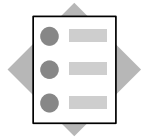
## Unit: Data Structures in Personnel Planning

- 1-1 The Personnel Planning data model is based on an **object-oriented design**.
  - 1-1-1 The following concepts form the basis of this design:
    - Object types
    - Relationships
    - Infotypes
  - 1-1-2 You can extend the Personnel Planning data model by defining new object types and new relationships in Customizing.
- 1-2 Additional data on relationships is made up of attributes that cannot be stored in isolation for the objects in question because they refer exclusively to the concrete relationship between two objects.
  - 1-2-1 The additional data is stored in the P1001 logical structure in the ADATA field.
- 1-3 A table infotype is an infotype whose data part has a repetitive structure.
  - 1-3-1 The PTnnnn structure contains the repetitive data fields for table infotypes.



## **Contents:**

- **Sequential and structural evaluations**
- **Evaluation paths**
- **Structural parameters and structural conditions**
- **Available internal information**



**At the conclusion of this unit, you will be able to:**

- **Explain the functions of the logical database PCH**
- **Create reports for sequential evaluations of the organizational structure**
- **Create reports for structural evaluations of the organizational structure**
- **Evaluate additional data on relationships**
- **Evaluate table infotypes**



Course Overview

Data Structures in  
Personnel Administration

The Logical Databases  
PNP/PNPCE

Join and Projection

Specific Commands

Repetitive Structures and  
List Display

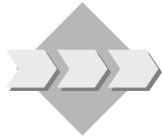
Cluster Database Tables in  
HR

Data Structures in  
Personnel Planning

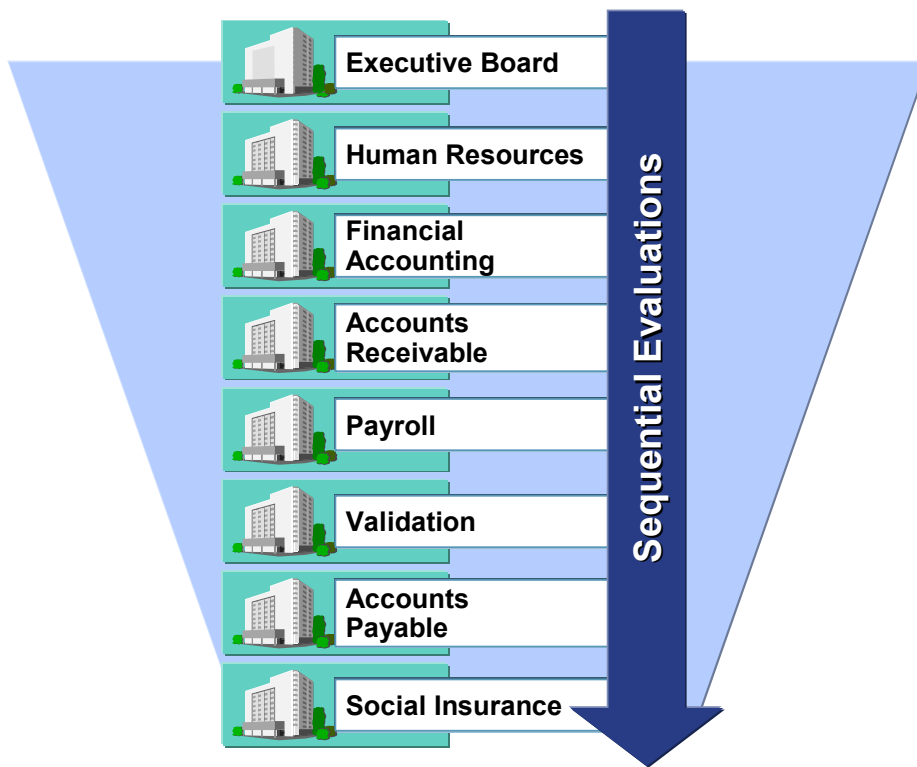
 **The Logical Database PCH**

Customer-Specific Infotypes

HR ABAP Features



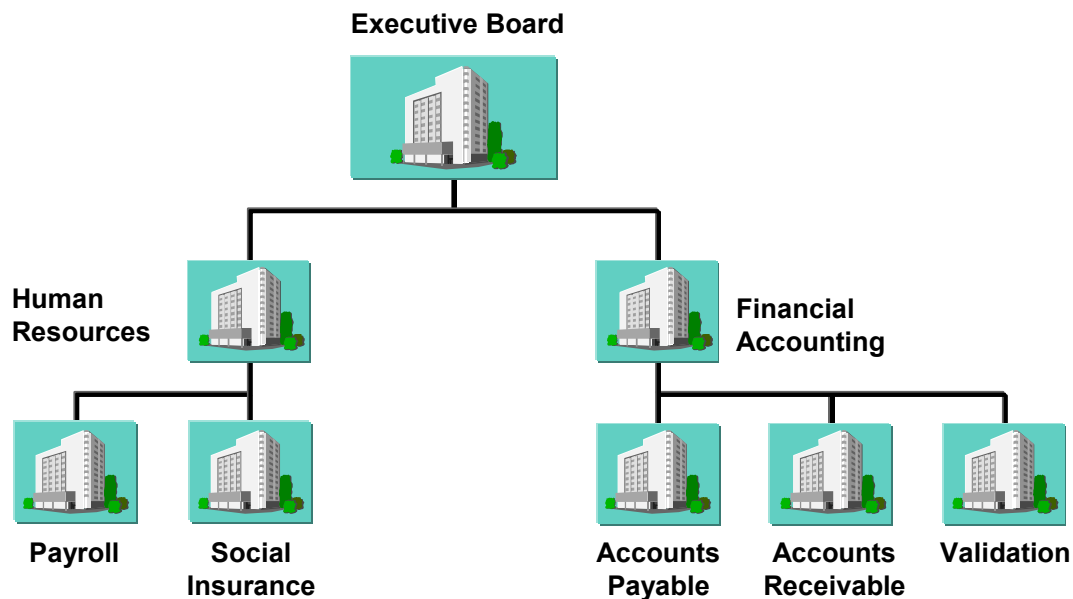
- **Your company's Executive Board requires an overview of the staff positions with the names of the employees holding these positions, based on the organizational structure.**



© SAP AG 2003

- You can specify objects for a sequential evaluation using their IDs. The sequential evaluation then takes place for all of the objects you have specified. For example, you can display a list of all organizational units at your company.

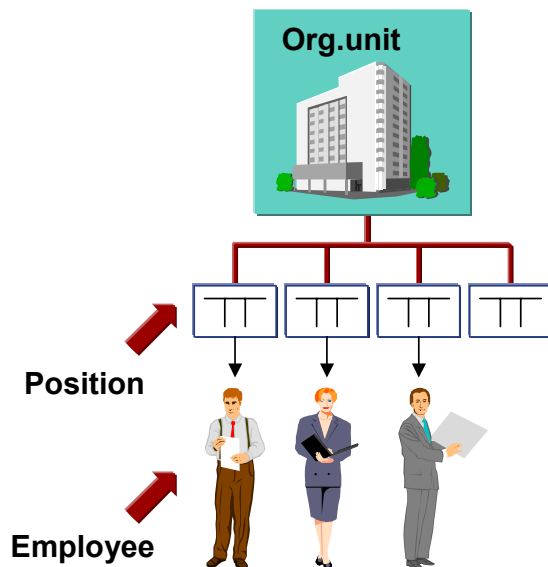
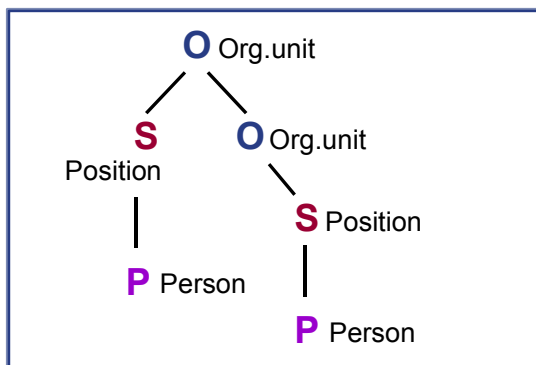
- When structural evaluations are performed, reports must take evaluation paths into account



© SAP AG 2003

- When a structural evaluation is performed, the objects to be evaluated are also listed. However, the system interprets these entries differently. It treats each selected object as a root object and uses it as a starting point for a hierarchical structure, which it builds up using a specific evaluation path.
- The evaluation path consists of a series of relationships to be evaluated, starting from the root object.
- Structural reports can be displayed using structural graphics.

- **Organizational units** O
- **Jobs** C
- **Positions** S
- **Persons** P



© SAP AG 2003

- An evaluation path describes a set of relationships between objects in a hierarchical structure. Evaluation path **O-S-P**, for example, describes the set of relationships found between organizational units, positions, and persons.
- Evaluation paths are used to select objects for structural evaluations. You choose an evaluation path, and the system evaluates the structure along the evaluation path. The report only evaluates objects that it finds in the specified evaluation path.
- Every standard report usually has a defined standard evaluation path. They are set in the system and must not be changed. The standard selection screen enables you to choose evaluation paths. You can also create new evaluation paths to meet your company's particular requirements.
- Report **RHWEGID0** displays all of the evaluation paths defined in the system, between the starting object type, and the target object type.

## Evaluation path O-S-P

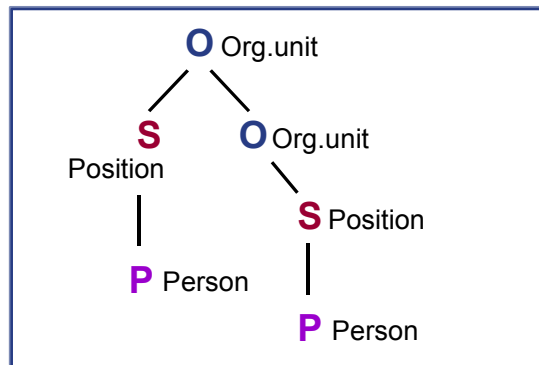
## Occupation of positions in line with organizational structure



No.	OT	A/B	Relationship	Priority	Type	Linked object	Skip
10	O	B	003 incorporates	*	S		
20	S	A	008 holder	*	P		
30	O	B	002 is line supervisor of	*	O		

### A/B

A = bottom up  
B = top down



© SAP AG 2003

- This evaluation path determines all of the assigned positions (S) and their holders (P) for a specific organizational unit (O). The subordinate organizational units are processed in exactly the same way.
- The "Skip" field enables you to determine that a specific relationship within an evaluation path is included in the evaluation but not displayed in the report list.
- Some evaluation paths consist of just one relationship. For example, **A001** is a subdivision of and **B001** is subdivided into. Thus, each relationship in the standard system has two evaluation paths.
- The convention A = *bottom up* and B = *top down* can be taken into account when a relationship is first defined. However, this is not an obligatory rule, and cannot be changed at a later stage.

```

TABLES:      OBJEC.
INFOTYPES:  0002,      "Personal data
            1003,      "Department/function
            ....
            nnnn.

GET OBJEC.
  LOOP AT Pnnnn
    WHERE BEGDA LE PC-ENDDDA
    AND   ENDDA GE PC-BEGDA.
    WRITE...
  ENDLOOP.
    
```

© SAP AG 2003

- To link a report with the **PCH** logical database, this database must be specified in the attributes.
- Use the TABLES statement to determine whether a sequential or structural evaluation is carried out. A sequential evaluation is carried out in the example outlined above.
- For each selected object, internal table Pnnnn is filled per infotype with all available infotype records. The internal tables can then be processed in a LOOP or PROVIDE.
- The infotypes from Personnel Administration can also be imported.
- In line with the PN structure of the PNP logical database, the PC structure is filled from the selection screen. PC-BEGDA and PC-ENDDDA contain the data selection period.
- When the GET event occurs, all information from the start of the object period for the infotype 1000 is in the **OBJEC** structure.

```
TABLES: OBJEC, GDSTR.  
INFOTYPES: 0002,          "Personal data  
            1003,          "Department/function  
            ....  
            nnnn.  
  
GET OBJEC.  
  LOOP AT Pnnnn  
    WHERE BEGDA LE PC-ENDDA  
    AND   ENDDA GE PC-BEGDA.  
    WRITE...  
  ENDLOOP.
```

© SAP AG 2003

- The only difference between a structural and sequential evaluation is the additional **GDSTR** entry in the TABLES statement. This ensures that the structure parameters are shown on the standard selection screen.
- **Note:**  
If you do not specify an evaluation path before starting the program, a sequential evaluation is performed.



### INITIALIZATION.

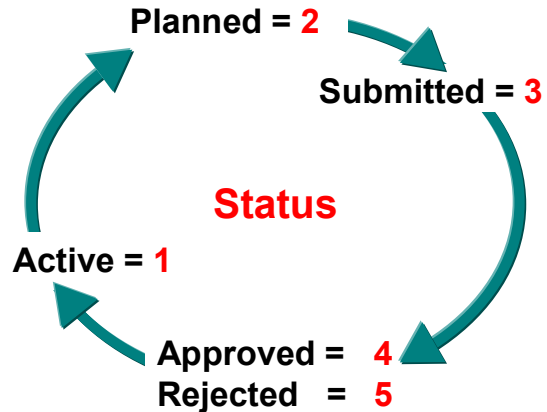
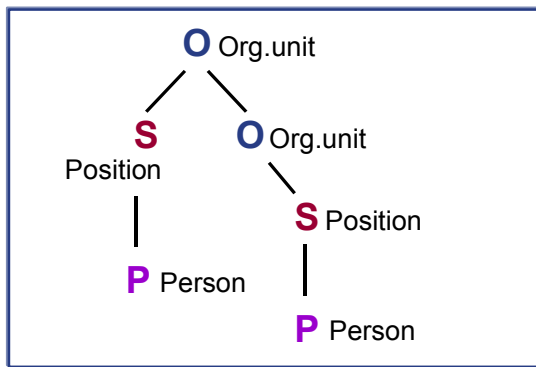
```
PCHPLVAR      = '01'.  
PCHOTYPE      = 'O'.  
PCHENDDA      = SY-DATUM.  
PCHWEGID      = 'O-S-P'.  
PCHOBJID-LOW  = '00000001'.  
APPEND PCHOBJID.
```

### GET OBJEC.

```
LOOP AT Pnnnn  
  WHERE BEGDA LE PC-ENDDA  
  AND   ENDDA GE PC-BEGDA.  
  WRITE...  
ENDLOOP.
```

© SAP AG 2003

- When the **INITIALIZATION** event occurs, you can set default values for the selection screen.
- The fields for the object ID are defined in **DBPCHSEL** include. This is an internal table (PCHOBJID) that must be filled with **APPEND**.
- You can use the **RH-SEL-ONE-OBJID** macro to suppress multiple selection for the Object ID field.
- You can use the **RH-SEL-KEYDATE** macro to restrict the evaluation period to one key day.

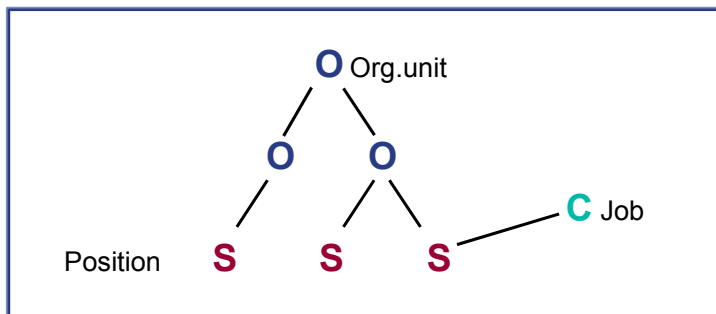


<b>Evaluation path</b>	<input type="text" value="O-S-P"/>	
<b>Status vector</b>	<input type="text" value="12"/>	<b>Status overlap</b>
<b>Display depth</b>	<input type="text" value="3"/>	
<b>Technical depth</b>	<input type="text"/>	<b>Recursion check</b>

© SAP AG 2003

- Enter the required evaluation path in the **Evaluation path** field.
- In the **Status vector** field, enter the status values required by relationship infotype 1001 along the evaluation path so that the appropriate target objects are selected. This parameter enables you to determine the objects irrespective of the status of the relationship infotypes along the evaluation path. For example, enter 12 (without a comma or blank character) to indicate that you only want to display objects whose relationships have status 1 "active" or status 2 "planned".
- The **Status overlap** checkbox is used in conjunction with the status vector field. This enables you to perform a simulation. The results are displayed once all of the relationships have been activated internally. During the simulation, all of the relationships are activated in accordance with the status specified in the Status vector field. Every status value as of position 2 is activated with the status value of position 1. For example, if the status vector is 123 and the status overlay parameter has been set, all relationships in status 2 and 3 are activated with status 1.
- The value entered in the **Display depth** field determines the hierarchical level up to which the structure is displayed. If the value of the field is 3, for example, the system only evaluates and displays the three highest hierarchical levels. In other words, the depth of the root object is 1.
- **Technical depth:** Depth at which a structure is read internally (interactive reporting).
- **Recursion check:** If the system detects recursion (for example, recursive data, recursive evaluation path), the system ends the selection.

<input checked="" type="radio"/> AND relationship	<input type="radio"/> Object filter		
<input type="radio"/> OR relationship	<input checked="" type="radio"/> Branch filter		
<b>CheckObjTy</b>	<b>Root object</b>	<b>Evaluation path</b>	<b>Status vector</b>
<b>S</b>	<b>C</b> 30015502	A007	



© SAP AG 2003

- You can set additional structure conditions that objects must meet. For example, you can evaluate all of the positions along the organizational hierarchy that are also described by one or more specific jobs.
- Read type for structure condition: Objects of the check object type must be accessible from the root object using the evaluation path.
- **AND relationship:**
  - All structure conditions must be fulfilled.
- **OR relationship:**
  - One of the structure conditions must be fulfilled.
- **Object filter:**
  - Irrelevant objects - that is, objects that do not meet the structure conditions - are hidden.
- **Branch filter:**
  - The entire branch below such objects is also hidden.

```
TABLES: OBJEC, GDSTR.
INFOTYPES: nnnn.

INITIALIZATION.
    PCHWEGID = 'SBES'.
. . .
GET OBJEC.
    LOOP AT Pnnnn
        WHERE BEGDA LE PC-ENDDA
        AND ENDDA GE PC-BEGDA.
        WRITE...
    ENDLOOP.
```

© SAP AG 2003

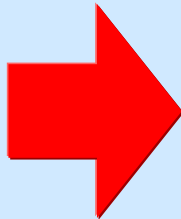
- Some structural evaluations only run with a fixed evaluation path that must not be overwritten on the selection screen.
- If you require a selection screen without structure parameters for such structural evaluations, enter **screen 900** in the report attributes.
- When structural evaluations take place using the sequential selection screen, the evaluation path must be defined in the report.
- The selection screen versions are stored in INCLUDE DBPCHSEL.

```
TABLES: OBJEC, GDSTR.  
INFOTYPES: nnnn.
```

```
. . .
```

```
GET OBJEC.  
  LOOP AT Pnnnn . . .
```

```
ENDLOOP.
```



**Pnnnn** Internal infotype tables

**OBJEC** Object information

**GDSTR** Root of structure

**STRUC** Structure information

© SAP AG 2003

- For each selected object, the internal table **Pnnnn** is filled per infotype with all available infotype records.
- For each selected object, the **OBJEC** structure contains key information from infotype 1000 (plan version, object type, object ID, short text, and long text).
- The **GDSTR** structure contains information on the current hierarchy, such as the root object and the number of objects within the hierarchy. The value of the structure does not change unless the system reaches a new hierarchy root when a structural evaluation is performed for more than one root object.
- For each selected object, the **STRUC** structure contains internal structure information. In particular, it includes information on the relationship used by the system to select the current object along the evaluation path.

- **Sequential evaluations**

```
INFOTYPES: 1001.      "Relationships
TABLES: OBJEC, PADnn.
. . .
GET OBJEC.
    LOOP AT P1001 WHERE RELAT = '0nn'.
        PADnn = P1001-ADATA.
    ENDLOOP.
```

- **Structural evaluations**

```
TABLES: OBJEC, GDSTR, PADnn, STRUC.
.
. . .
GET OBJEC.
    PADnn = STRUC-VADATA.
or:
    IF STRUC-VRELAT = '0nn'.
        PADnn = STRUC-VADATA.
    ENDIF.
```

© SAP AG 2003

- You must ensure that the relationship type is queried (in accordance with table T77AR) before an assignment is made to an additional data structure that is dependent on the relationship type.
- The P1001-ADATA field contains the additional data of the relationships for P1001, that is, the relationships to the current object.
- The STRUC-VADATA field contains the additional data for the current relationship, that is, the relationship that was used to reach the current object.

```

INFOTYPES: 1002.                "Description
TABLES: OBJEC.
DATA: ipt1002 LIKE PT1002 OCCURS 0 WITH HEADER LINE.
. . .

GET OBJEC.
  WRITE: / OBJEC-OTYPE, OBJEC-OBJID.
  LOOP AT P1002.
    RH-GET-TBDAT P1002-INFTY P1002-TABNR ipt1002.
    LOOP AT ipt1002.
      WRITE: / ipt1002-TLINE.
    ENDLOOP.
  ENDLOOP.

```

© SAP AG 2003

- RH-GET-TBDAT is a macro for the logical database PCH. These macros are defined in the include DBPCHCOM. You can use this macro to import the data for an infotype with repetitive structures. For more information, see the relevant pages on **Table Infotypes**.
- The parameters for the macro RH-GET-TBDAT are:
  - Parameter 1 : Infotype
  - Parameter 2 : Reference field
  - Parameter 3 : Table for the PTnnnn structure
- Do not confuse macros for the logical database PCH with macros for the logical database PNP. Note in particular that you cannot use macros for the logical databases PCH and PNP at the same time.

```

INFOTYPES: 1003.                "Department/function
TABLES: OBJEC.

START-OF-SELECTION.
  RH-SET-INDEX-INFTY '1003'.
  RH-CONDITION-LINE ABTEL EQ 'X' SPACE.
  RH-SET-INDEX-INFTY-CONDITION.
  . . .

GET OBJEC.
  WRITE: / OBJEC-OTYPE, OBJEC-OBJID.
  LOOP AT P1003 WHERE ABTEL NE SPACE.
    WRITE: /3 P1003-BEGDA, P1003-ENDDA.
  ENDLOOP.
    
```

© SAP AG 2003

- The call sequence of PCH macros facilitates fast object selection using value conditions of infotype fields (infotype index). You must use this call sequence when objects are selected for sequential evaluations on the basis of whether infotypes exist with specific field values, rather than using the object ID.
- The **RH-SET-INDEX-INFTY** macro specifies the infotype to be indexed.
- The parameters of the **RH-CONDITION-LINE** macro are:
  - Parameter 1 : Field name (for example, ABTEL)
  - Parameter 2 : Condition (for example, EQ, BT)
  - Parameter 3 : Value (for example, "X")
  - Parameter 4 : Value (for Between)
- The **RH-SET-INDEX-INFTY-CONDITION** macro is used to set the condition as an index.
- You can use this method for sequential evaluations, but not for structural evaluations.

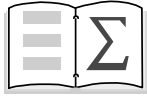


```

INFOTYPES: 1001.                "Relationships
DATA: ihobject LIKE HROBJECT OCCURS 10.
. . .
CALL FUNCTION 'RH_READ_INFITY'
EXPORTING
. . .
    INFTY                = '1001'
    BEGDA                = <begdat>
    ENDDA                = <enddat>
TABLES
    INNNN                = P1001
    OBJECTS              = ihobject
EXCEPTIONS
    ALL_INFITY_WITH_SUBTY = 1
    NOTHING_FOUND        = 2
    NO_OBJECTS           = 3
    WRONG_CONDITION      = 4
    OTHERS               = 5.
    
```

© SAP AG 2003

- The RH\_READ\_INFITY function module can be used to read Personnel Planning infotype records (function group RHDB) in programs without the PCH logical database (module pools, reports of other logical databases).
- The function module reads all of the infotype records for a specific infotype or (parameter INFTY = SPACE) all of the infotypes for the specified set of objects.
- Depending on the AUTHORITY and WITH\_STRU\_AUTH parameters, the function module performs an authorization check.



**You are now able to:**

- **Explain the functions of the logical database PCH**
- **Create reports for sequential evaluations of the organizational structure**
- **Create reports for structural evaluations of the organizational structure**
- **Evaluate additional data on relationships**
- **Evaluate table infotypes**



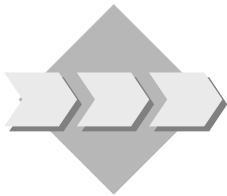
## Unit: The PCH Logical Database

### Topic: Creating a List of Staff Positions



At the end of this exercise, you will be able to:

- Program a structural evaluation
- Select administration infotypes using the PCH logical database
- Understand the search in depth evaluation principle



Your company's Board of Directors requires an overview of the staff positions with the name of the employees holding the positions, and their telephone number.

- 1-1 Write a report that lists the staff positions along the organizational structure, and the names of the employees holding the positions.  
Use the **O-S-P** evaluation path.
  - 1-1-1 Take the personal data (last name, first name) from infotype 0002, and take the telephone number from infotype 0006 subtype 1.
  - 1-1-2 The information on the staff positions is contained in the position in infotype 1003. If a position is marked as a staff position, the value "X" is selected for the **P1003-STABS** field.
  - 1-1-3 Process infotypes 0002 and 0006 of Personnel Administration using a join. For the infotypes, project to the fields to be issued: First name, last name, telephone number, and subtype.
  - 1-1-4 In one row, **only** enter the object ID, the object abbreviation and the object name for the employees holding the positions. In the following row, enter the name and telephone number.
  - 1-1-5 The plan version **01**, object type **O**, object ID **30014999** and evaluation path **O-S-P** should appear as default values.



The PCH logical database displays the object in the following sequence: Organizational unit – Position – Person. The **P1003** internal table is only filled for the object **S** at the time the GET OBJEC event occurs. However, the table is blank again for the following object **P** Person. Therefore, select the appropriate indicator if the position for the person is a staff position.





## Unit: The PCH Logical Database

### Topic: Creating a List of Staff Positions

```
REPORT zhsol010.
TABLES: objec, gdstr.
INFOTYPES: 0002, 0006, 1003.

DATA: stabs LIKE p1003-stabs,
      name LIKE pernr-ename.

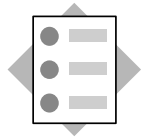
INITIALIZATION.
  pchplvar = '01'.
  pchotype = 'O'.
  pchwegid = 'O-S-P'.
  pchobjid-low = '30014999'.
  append pchobjid.

GET objec.
  IF objec-otype = 'S'.
    CLEAR stabs.
    LOOP AT p1003 WHERE begda LE pc-endda
          AND endda GE pc-begda.
      IF p1003-stabs = 'X'.
        stabs = 'X'.
        WRITE : / objec-objid, objec-short, objec-stext.
      ENDIF.
    ENDLOOP.
  ENDIF.
  IF objec-otype = 'P' AND stabs = 'X'.
    PROVIDE vorna nachn FROM p0002
      subty telnr FROM p0006
      BETWEEN pc-begda and pc-endda
      WHERE p0006-subty = '1'.
    IF p0006_valid = 'X'.
      CONCATENATE p0002-vorna p0002-nachn INTO name
      SEPARATED BY SPACE.
      WRITE: / name, p0006-telnr.
      SKIP.
    ENDIF.
  ENDPROVIDE.
ENDIF.
```



### **Contents:**

- **Enhancement of infotypes**
- **Creating infotypes in Personnel Administration**
- **Creating infotypes in Personnel Planning**



**At the conclusion of this unit, you will be able to:**

- **Add customer-specific fields to standard infotypes**
- **Create a customer-specific infotype in Personnel Administration**
- **Create a customer-specific infotype in Personnel Planning**



Course Overview

Data Structures in  
Personnel Administration

The Logical Databases  
PNP/PNPCE

Join and Projection

Specific Commands

Repetitive Structures and  
List Display

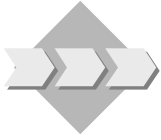
Cluster Database Tables in  
HR

Data Structures in  
Personnel Planning

The Logical Database PCH

 Customer-Specific Infotypes

HR ABAP Features



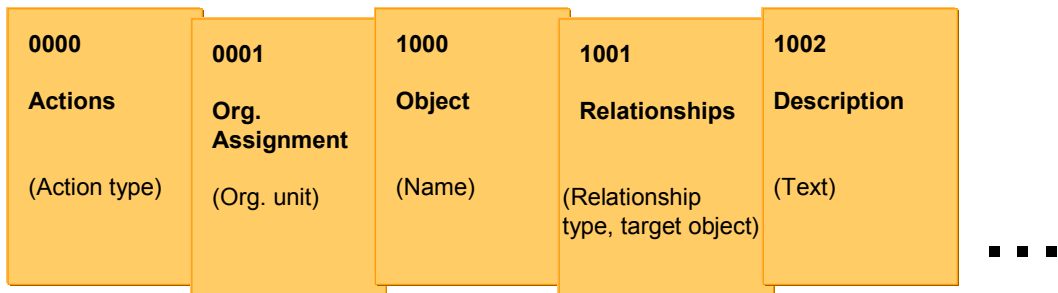
- Your company has data that cannot be stored in a standard infotype. You want to save this data in customer-specific infotypes.

- "What is an infotype?"

- Structural description in the Repository
- Database table
- Editing program (module pool)
- Screens 1000, 2000, 3000
- Table entries for control

Key fields, general fields, PSnnnn and HRInnnn  
 PAnnnn and HRPnnnn  
 MPnnnn00  
 MPnnnn00 / 1000, 2000, 3000  
 for example, T582A, T582S, T777I, and T777Z

Standard infotypes:

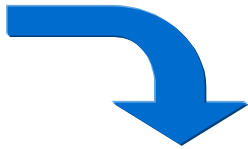


© SAP AG 2003

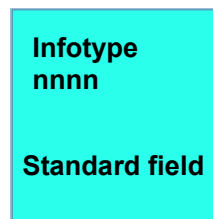
- Each infotype has an entry in the check table (T777D), and several infotype-dependent components in the Repository.
- In addition to the structures and database tables mentioned in Unit 2 (Data Structures in Personnel Administration), each infotype has a module pool that contains an infotype-specific entry and a list screen with validations, and a dialog module.
- When you create the individual components, you must note that:
  - Specific infotype-dependent objects of infotypes must be edited using the Data Dictionary.
  - You should edit certain components of infotypes only using the specially designed Personnel Planning and Personnel Administration infotype copier.
  - You should make certain infotype-specific settings using infotype Customizing.

If you take account of these notes, you will avoid inconsistencies in infotype-specific objects and table entries. More details on creating and enhancing infotypes are provided in the rest of this unit.

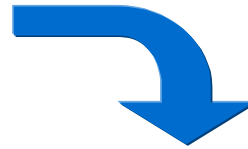
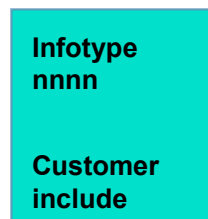
- Where can company-specific information be stored?



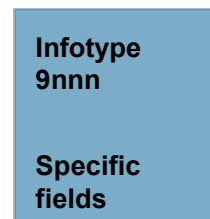
In standard fields of  
standard infotypes



In the customer include  
of standard infotypes



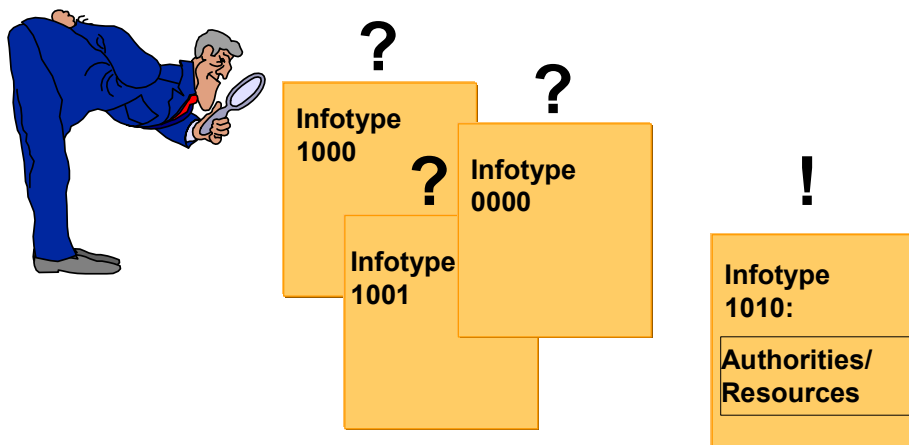
In (new) customer  
infotypes



© SAP AG 2003

- You have three options for creating customer-specific information in infotypes:
  - You can use a standard infotype with the fields available. For example, you can store the desired information in a specific field of a standard infotype.
  - You can enhance a standard infotype. That is, you can create the fields you require in a customer include.
  - You can create a new infotype with the fields you require.
- Before you enhance standard infotypes or create new infotypes, always check whether the infotypes available in the standard system meet your requirements.

- When can you use standard infotypes?

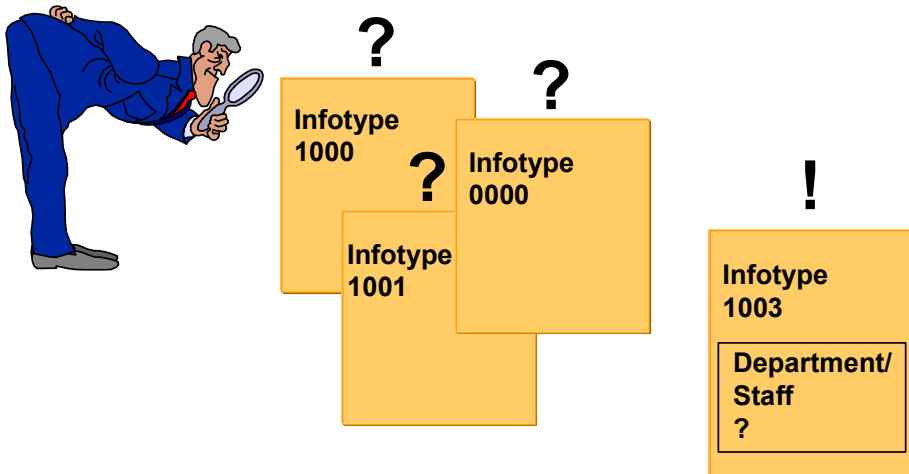


- When a suitable field exists in the standard infotypes

© SAP AG 2003

- Problem: You want to store specific information in the system.
- Procedure: Check all standard infotypes to see if one of the standard infotype fields can be used to store this information.
- Situation: You find an infotype that contains a suitable field.
- Procedure: Use this infotype to store your specific information.
- For example: You would like to store the information “Position entitled to company car”. You can use the standard infotype 1010 (Authorities/Resources) to do this.
- Note: You can retrieve information on the specific fields from the infotype-specific structure Pnnnn (or from the PSnnnn or HRInnnn structures).

- When can you use enhanced standard infotypes?

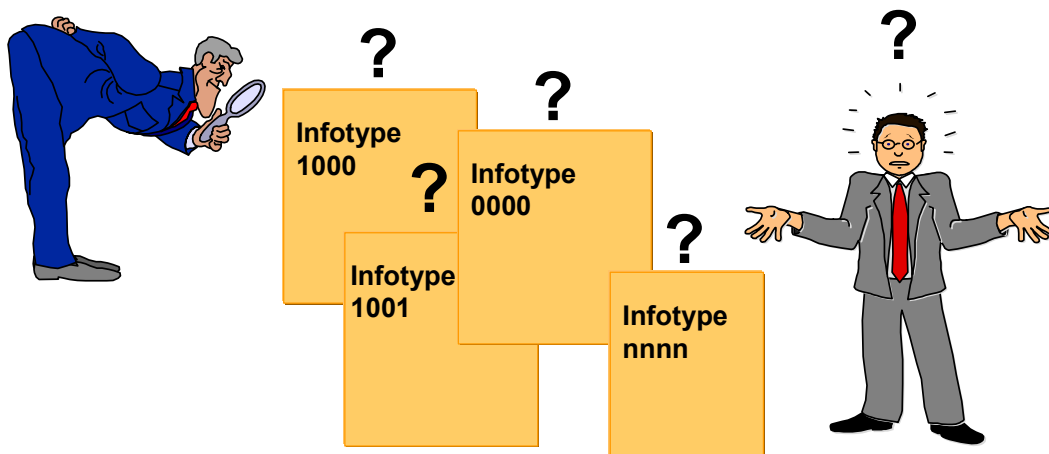


- When a standard infotype meets most of your requirements

© SAP AG 2003

- Problem: You want to store specific information in the system.
- Procedure: Check all standard infotypes to see if one of the standard infotype fields can be used to store this information.
- Situation: You find an infotype that is suitable for most of your data. Additional fields are required, however.
- Procedure: Use the customer include structure for this infotype (CI\_Pnnnn) to store specific information by adding the required fields to this structure.
- Example: In an organizational unit, you would like to store the information that this organizational unit is managing a project.  
You use the customer include of the standard infotype 1003 (*Department/Staff*), and add the *Project Management* field.
- Advantage: The enhanced infotype does not change the infotype already defined in the standard version. In particular, no database conversions are required during upgrades, if SAP simultaneously adds enhancements to this infotype in the standard version (different field sequence possible between Dictionary and Database).  
In addition, evaluations only require minimal change.

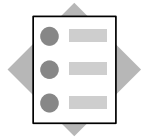
- When can you use customer-specific infotypes?



- When you cannot store the required information in any of the standard infotypes

© SAP AG 2003

- Problem: You want to store specific information in the system.
- Procedure: Check all standard infotypes to see if the fields of one of the standard infotypes can be used to store this information.
- Situation: The information cannot be stored in any of the standard infotype fields.
- Procedure: Use one of the infotype copiers described in the following sections to create a new infotype that meets your specific requirements.



**At the conclusion of this topic, you will be able to:**

- **Add customer-specific fields to a Personnel Administration infotype**



**Contract Elements (infotype 0016)**

**Contractual regulations**

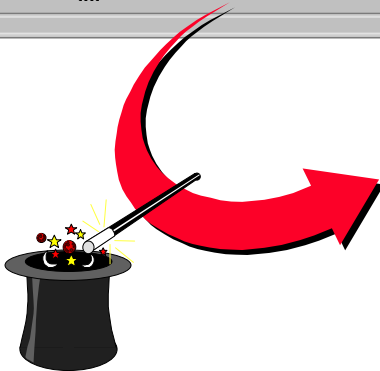
Contract type  Unlimited  
.... ....

**Payment period from beginning of illness**

Continued pay  Days  
Sick pay supplement  Year

**Periods**

Probationary  Months  
.... ....



© SAP AG 2003

**Contract Elements (infotype 0016)**

**Contractual regulations**

Contract type  Unlimited  
.... ....

**Payment period from beginning of illness**

Continued pay  Days  
Sick pay supplement  Year

**Periods**

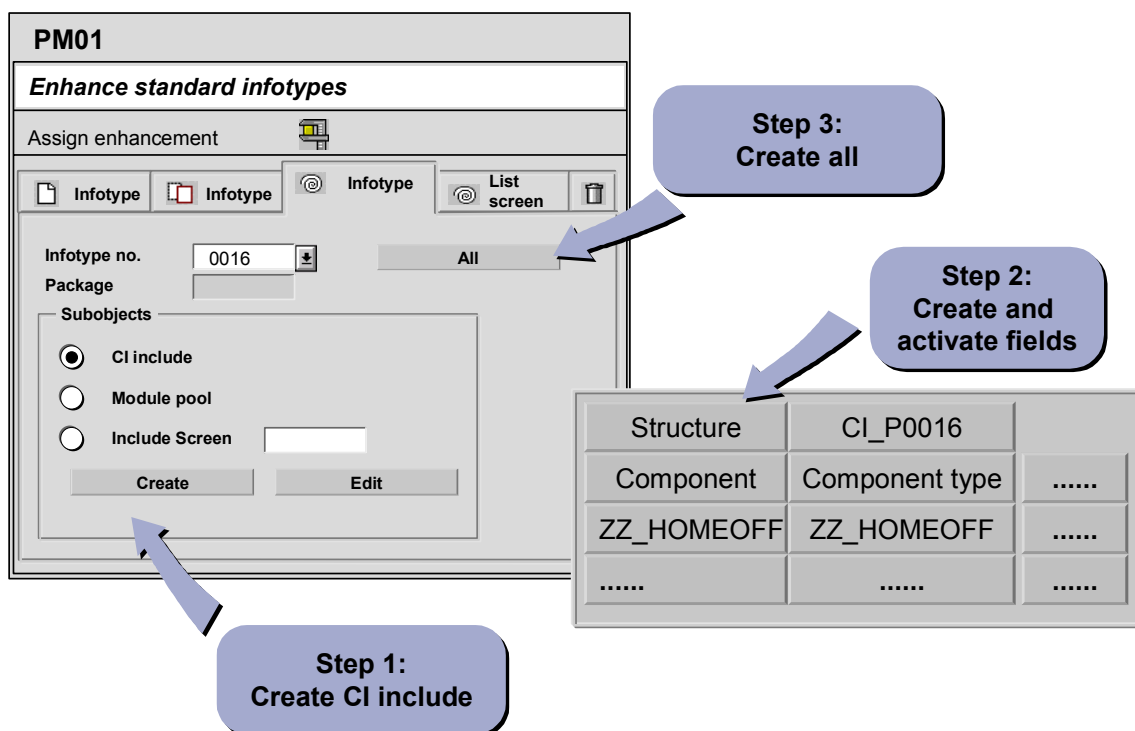
Probationary  Months  
.... ....

**Additional fields**

Customer-specific 1   
Customer-specific 2   
Customer-specific 3



- You can use the enhancement concept to add any number of new fields to a Personnel Administration, Recruitment or Personnel Planning infotype.
- When you add new fields to an infotype, these fields are handled as standard SAP fields in dynamic actions and when creating infotype logs.
- Note that enhancements are not classed as modifications. Adding fields to an infotype in the SAP standard system does not cause any problems during an upgrade.
- The following infotypes are among those not covered by the enhancement concept:
  - Actions infotype (0000)
  - Additional Actions infotype (0302)
  - Time Management infotypes (2nnn)
  - Object infotype (1000)
  - Relationships infotype (1001)



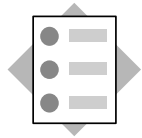
© SAP AG 2003

- Start the infotype copier in Personnel Administration (transaction PM01). The *Create Infotype* screen appears. Select *Expand Infotype*. In the *Infotype Number* field, enter the four-digit number of the infotype you would like to enhance. When you enter the infotype number, remember to enter leading zeros as well.
- In the *Subobjects* groupbox, select *CI include*, and choose *Create*. The screen *Dictionary: Initial Screen* appears.
- Create the CI include and choose *Activate*. If you would like to add completely new fields, you must also create data elements. Go back to the *Enhance Infotypes* screen.
- Select *Create All*.
- Result: You have added additional fields to the standard single screen of an infotype.

- **CI\_Include**      **C\_Pnnnn**      **Customer-specific data fields**
- **Module pool**      **ZPnnnn00**
- **Include screen**      **ZPnnnn00 0200**
- **Includes**      **ZPnnnn10**      **Declaration of joint data objects**  
                          **ZPnnnn20**      **PBO module for screens**  
                          **ZPnnnn30**      **PAI module for screens**  
                          **ZPnnnn40**      **Subroutines**

© SAP AG 2003

- The include screen is assigned to the standard screen. To do this, use the *Assign enhancement* button. The entries in table T582C ensure that the screen underneath the standard fields is displayed.
- Calling a subscreen for the processing times PBO and PAI: CALL SUBSCREEN SUBSCREEN\_T582C cause the customer enhancement to be taken into account in the module pool of the standard screen.



**At the conclusion of this topic, you will be able to:**

- **Create a customer-specific infotype in Personnel Administration**

**PM01**

**Create Infotype**

Infotype characteristics    Technical attributes   

☐ Infotype

Infotype no.    9nnn    All

Package   

☐ Employee infotype    ☐ Applicant infotype    ☐ Both

Subobjects

☒ PS structure

☐ Module pool

☐ User screen   

☐ Interface

**1. Create and activate PS9nnn structure**



**Step 2:**

© SAP AG 2003

- Step 1: Call the Personnel Administration infotype copier (transaction PM01).
- You must store the infotype-specific fields for your infotype in the system. To do this, create the structure PS9nnn in the Dictionary.
  - Save and activate the structure.
  - You may need new data elements and domains for your structure.

**PM01**

**Create Infotype**

Infotype characteristics Technical attributes

Infotype

Infotype no. 9nnn

Package

☐ Employee infotype ☐ Applicant infotype ☐ Both

Subobjects

☒ PS structure

☐ Module pool

☐ Screen

☐ User Interface

Step 1:

2. Create and activate all objects

■ Step 2: Once you have created the PS9nnn structure, choose *Create All* in transaction PM01.

■ You now create a new infotype.

- When you do this, the infotype copier automatically creates the following objects:
  - Database table PA9nnn
  - Structure P9nnn
  - Module pool: MP9nnn00
  - 3 screens (1000, 2000, 3000) for the module pool
  - Interface status
  - Dialog module (RP\_9nnn)
- When you create new infotypes, the following table entries are also automatically created:
  - T777D Infotype check table
  - TDCD Dialog modules

## Creating Personnel Administration Infotypes - Step 3

SAP

<b>Infotype</b>		<b>9001</b>	<b>Company carpark</b>
<b>General attributes</b>			
Time constraint	<b>2</b>	<input type="checkbox"/> Subtype obligatory	<input type="checkbox"/> Accntng/log.data
Time cnstr. tab.		Subtype table	<input checked="" type="checkbox"/> Text allowed
Maint.aft.leave	<input type="checkbox"/>	Subty. text tab.	<input type="checkbox"/> Copy infotype
<input checked="" type="checkbox"/> Access auth.		Subtype field	<input type="checkbox"/> Propose infotype
<b>Display and selection</b>			
Select w/start	<b>3</b>	Create w/o strt	<b>1</b>
Select w/ end	<b>5</b>	Create w/o end	<b>1</b>
Select.w/o date	<b>2</b>	<input type="checkbox"/> List time per.	<input checked="" type="checkbox"/> Choose data
<b>Retroactive accounting trigger</b>			
<input type="checkbox"/> Before ERA date	<input type="checkbox"/> Entry of RA limit time	Past entry all. <b>x</b>	
Retr.acct.payr.		Retr.acct.PDC	No org.assign.
<b>Technical data</b>			
Single screen	<b>2000</b>	Dialog module	<b>RP_9001</b>
List screen	<b>3000</b>	Structure	<b>P9001</b>
<input type="checkbox"/> List entry		Database table	<b>PA9001</b>
		Applicant infotypes	
		Applicant DBTab	

- When you create an infotype, the system does not automatically create the table entries that describe the infotype's characteristics (tables T582A and T582S). You must therefore create the corresponding entries manually in the tables. You use the view V\_T582A to maintain these tables. This view appears after you carry out the *Create All* step.
- You can maintain these tables at any time in transaction PM01 on the *Create infotype* screen. To do this, select *Infotype characteristics*.
- You can create new entries by:
  - Selecting *New Entries*, or
  - Copying an entry for an infotype with similar characteristics

- **Structures P9nnn and PS9nnn**
- **Database table PA9nnn**
- **Module pool MP9nnn00 with includes**
  - MP9nnn10 (data declaration)**
  - MP9nnn20 (PBO modules for screens)**
  - MP9nnn30 (PAI modules for screens)**
  - MP9nnn40 (Subroutines)**
- **Screens 1000, 2000, and 3000 for module pool**
- **GUI status with standard icons**
- **Dialog module RP\_9nnn**
- **Entries in T777D check table infotypes and TDCD dialog modules**

© SAP AG 2003

- Select *Create All* in transaction PM01 to create the objects outlined above.
- In the screens, you often have to modify the *Layout* and *Flow logic* components.
- You usually do not have to make any further modifications to the GUI status.



```

PROGRAM MP9nnn00 MESSAGE-ID RP.

TABLES: P9nnn.
* the following tables are filled globally:
* T001P, T500P
* they can be made available with a TABLES statement

FIELD-SYMBOLS: <PNNNN> STRUCTURE P9nnn
                DEFAULT P9nnn.

DATA: PSAVE LIKE P9nnn.
    
```

© SAP AG 2003

- Global data is created in the MP9nnn10 include. These are the declarations for the tables and work fields used.

```
MODULE P9nnn OUTPUT.  
  IF PSYST-NSELC EQ YES.  
    * read text fields etc.; do this whenever the screen is  
    * shown for the first time:  
    *   PERFORM RExxxx.  
    IF PSYST-IINIT = YES AND PSYST-IOPER = INSERT.  
    * generate default values; do this the very first time  
    * on insert only:  
    *   PERFORM GET_DEFAULT.  
    ENDIF.  
  ENDIF.  
ENDMODULE.
```

© SAP AG 2003

- The MP9nnn20 include contains the modules that are called when the PBO event occurs. To allow the program to be structured better in subroutines, the processing of the data is defined in the include MP9nnn40.
- To control processing, it is useful to query specific values of the **PSYST** structure.
- The **NSELC** indicator has the value YES (= 1), if a record is displayed on screen for the first time. It should be queried, so that it is only read once when the PBP text for keys is run repeatedly.
- The **IINIT** indicator has the value YES, if a record is displayed for the first time. It is used to carry out special processing actions of the PBO for this record, for example, inserting default values when adding elements.
- The only difference between these indicators is that when you go to the overview when processing a record, IINIT is still on NO (= 0) after you go back to processing. This prevents default values that are changed by the user from being renewed. If a user overwrites the default values during creation, and goes to the overview during entry, the variable IINIT is at NO when you return to the entry screen. The default values are not to be redetermined. However, the NSELC variable has the value YES as the system must read the texts again.

```
PROCESS BEFORE OUTPUT.  
*      General infotype-independent processing  
MODULE BEFORE_OUTPUT.  
  CALL SUBSCREEN subscreen_empl  
    INCLUDING empl_prog empl_dynnr.  
  CALL SUBSCREEN subscreen_header  
    INCLUDING header_prog header_dynnr.  
*      Infotype-specific processing  
MODULE P9nnn.  
*  
MODULE HIDDEN_DATA.
```

© SAP AG 2003

- You can carry out infotype-specific initializations in the **P9nnn** PBO module. Here, you can, for example, fill the screen fields that are stored in the **Qnnnn** (SAP) and **Znnnn** (customer) structures.
- Do not change the **BEFORE\_OUTPUT** and **HIDDEN\_DATA** PBO modules.

```
PROCESS AFTER INPUT.  
*   Exit processing.  
    MODULE EXIT AT EXIT-COMMAND.  
*   Processing after input  
*   Note when entry has been made: All screen fields that  
*   are ready for input must be listed here  
  
CHAIN.  
    FIELD P9nnn-BEGDA.  
    FIELD P9nnn-ENDDA.  
    FIELD P9nnn-field1,...  
    MODULE INPUT_STATUS ON CHAIN-REQUEST.  
ENDCHAIN.
```

© SAP AG 2003

- The **INPUT\_STATUS** PAI module must be executed if the user makes an entry in a screen field. Therefore, all entry fields must be listed in the chain.

```

*   Function code editing before entry checks
MODULE PRE_INPUT_CHECKS.
*   Entry checks:
*   Insert check modules here:
    FIELD P9nnn-field1 ON INPUT MODULE CHECK_FIELD1.
*   Processing of function code entry: ALL other fields
*   that appear on the screen must be listed here, even
*   the only output fields
CHAIN.
    FIELD P9001-BEGDA.
    FIELD P9001-ENDDA.
    FIELD RP50M-SPRTX.
    FIELD P9nnn-field1,...
    MODULE POST_INPUT_CHECKS.
ENDCHAIN.
    
```

© SAP AG 2003

- The function code is processed before the entry check in the **PRE\_INPUT\_CHECKS** PAI module. The system exits processing of the current individual screen if, for example, you select *Exit*.
- Once you have processed the **PRE\_INPUT\_CHECKS** module, you can make your own entry checks or call your own PAI modules. In the above example, an entry in the *P9nnn-field1* field is checked by the **CHECK\_FIELD1** module.
- The entry checks must be completed up to the **POST\_INPUT\_CHECKS** PAI module. It is not possible to change field contents after subsequent processing.
- The function code is processed after the entry check in the **POST\_INPUT\_CHECKS** PAI module. General entry checks are also carried out here. For the general entry checks, the system checks for example, whether the start date of the infotype record is before the record's end date.
- All fields that appear on the single screen must be listed in the chain. Fields that are only displayed, for example, long texts, are also to be listed here.

Screen field is ready-for-input for the function	Hexadecimal constant for modification group 1
<i>Display</i>	<b>001</b>
<i>Change</i>	<b>002</b>
<i>Add and copy</i>	<b>004</b>
<i>Delete</i>	<b>008</b>
<i>Lock/Unlock</i>	<b>010</b>

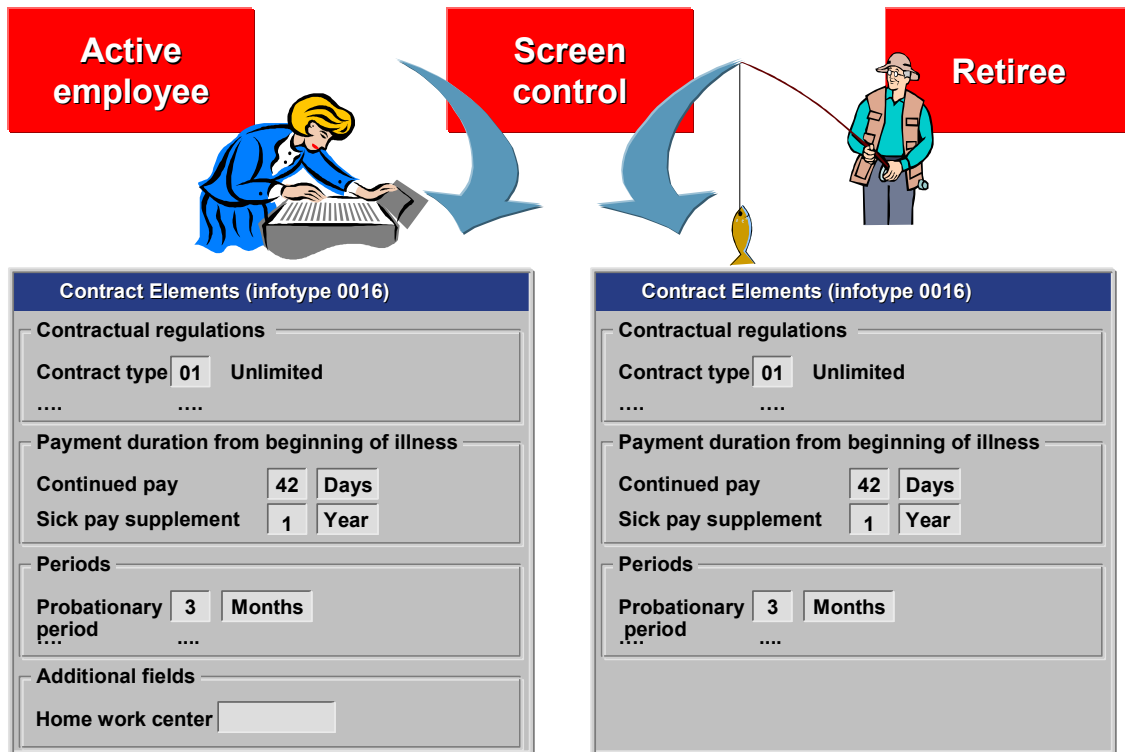
© SAP AG 2003

- You determine the attributes of the individual screen fields when you create the single screens and list screens with the ABAP Screen Painter. However, the same screen is always used for different functions such as creating, displaying, maintaining, and deleting infotype records. Therefore, when you maintain the screen, you cannot specify whether or not a screen field is to be ready for input. Depending on the employee's organizational data, certain screen fields may need to be hidden.
- Some attributes can only be specified at runtime, and cannot be specified in general. The screens' layout is therefore to change, depending on the function a user has chosen, or on the data to be edited.
- Controlling whether screen fields are ready for input or hidden:
- You can use the *Modification group 1* value to determine whether screen fields are ready for input, depending on the function to be carried out. You can also hide specific screen fields.
- The value of *Modification group 1* must be maintained for all input fields.
- A screen field is not ready for input if you have not maintained this entry.

Screen field is hidden for the function	Hexadecimal constant for modification group 1
<i>Delimit in list screen</i>	<b>200</b>
<i>Display in list screen and change in list screen</i>	<b>400</b>
<i>Add and copy</i>	<b>800</b>

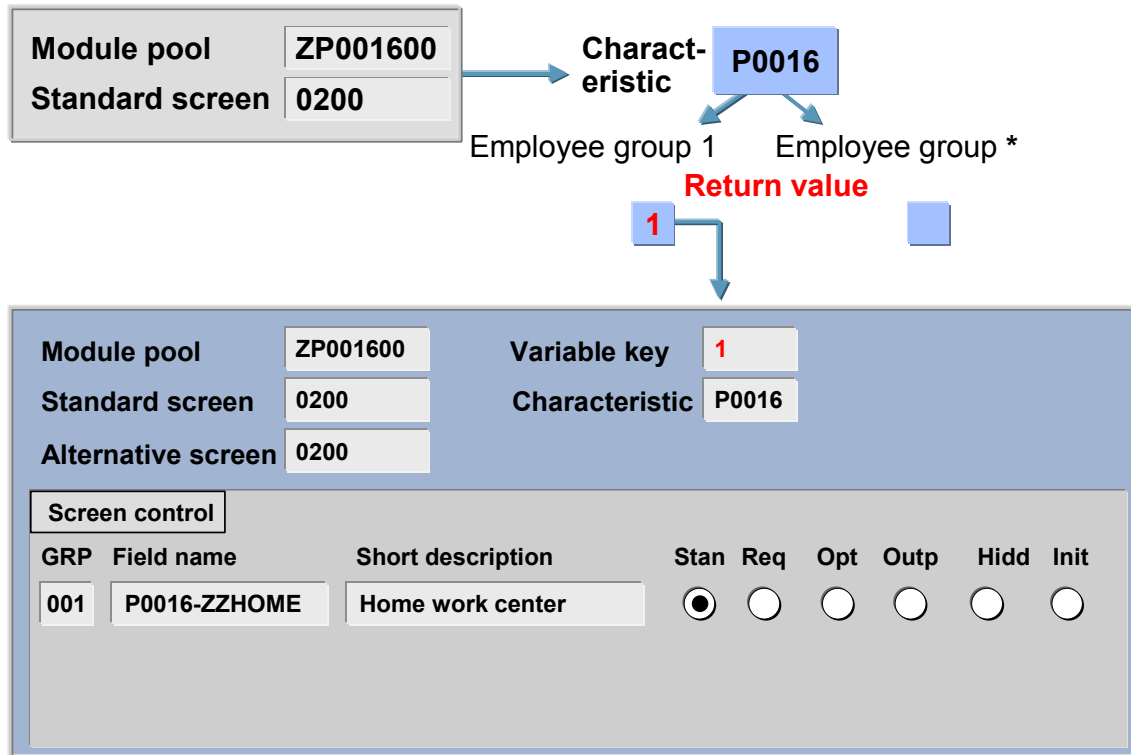
© SAP AG 2003

- The value of *Modification group 1* is interpreted bit-by-bit. Several constants can be combined with each other. For this reason, the values are added. Note that the *Modification group 1* value must be maintained in hexadecimal form.
- For example:  
You would like a screen field to be ready for input for the *Add* and *Change* functions. You therefore maintain the value **006** in *Modification group 1*. You would like one screen field to be ready for input for all functions. Here, you maintain the value **00F** in *Modification group 1*.
- The **BEGDA** and **ENDDA** fields are usually ready for input for all actions except for the display of records. You therefore assign the value **00E** to the *Modification group 1* attribute of these fields.
- *Modification group 1* has the value **800** for the *AEDTM* and *UNAME* fields. This means that these fields are hidden when you add a record.



- Together with control data, you can use screen control to:
  - Replace the standard screen with an alternative screen
  - Control whether specific screen fields are ready for input
  - Hide specific screen fields
- In general, screen control can be carried out based on organizational data for the employee or based on the infotype record's subtype. To this end, maintain the value of *Modification group 3* in the ABAP Screen Painter for the corresponding screen fields.
- Each screen field is assigned a value between **001** and **050** in *Modification group 3*. Use the same value for screen fields that are to be modified in the same way. The same value is used for an input/output field as for the corresponding keyword and any long text that is displayed. Assign screen fields that are not modified using T588M to the value *SPACE* in *Modification group 3*.





© SAP AG 2003

- In table T588M, you then determine which alternative screens you would like to use, if any, and how the individual screen fields are to be modified.
- If you require a modified screen, but do not find any entry in this table, create a new basic entry for the screen and make sure that the *Variable key* field is empty.
- If the screen modification is to apply for the infotype in general, you do not require any characteristic or any other entries in this table.
- As is the case in this example, the screen modification can be based on the employee's organizational assignment. The modification is determined by the Pnnnn characteristic (nnnn = infotype number). If you do not enter a return value in the characteristic's decision tree, the entry for which the *Variable key* field is empty, that is the basic entry (basic screen), is automatically valid. If no characteristic is available for the corresponding infotype, you must create one.
- Create a new entry with the return value in the variable key for each return value for the characteristic. If necessary, assign the screen number of an alternative screen. The screen modification in the detail view is now also valid for the alternative screen.



**At the conclusion of this topic, you will be able to:**

- **Add customer-specific fields to a Personnel Planning infotype**

**PPCI**

**Enhance Infotype**

Infotype: 1003  
Infotype name: \_\_\_\_\_

Single screen:  
☒ CI include All  
☐ Module pool  
☐ Include screen 200

Create Change

**Step 1: Create CI include**

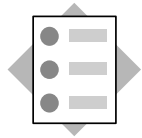
Structure	CI_P1003	
Component	Component type	.....
ZZ_PROJECT	ZZ_PROJECT	.....
.....	.....	.....

**Step 2: Create and activate fields**

**Step 3: Create All**

© SAP AG 2003

- Start the Personnel Planning infotype copier (transaction PPCI). In the *Infotype* field, enter the four-digit number of the infotype you would like to enhance. Select *Extend*.
- The *Enhance Infotype* screen appears. Select *Create All*. The *Dictionary: Initial Screen* appears.
- Create the include and choose *Activate*. If you would like to add completely new fields, you must also create data elements. Go back to the *Enhance Infotype* screen.
- Result: You have added additional fields to the standard single screen of an infotype.
- **Note:**
  - It is not possible to enhance the infotypes 1000 (*Object*) or 1001 (*Relationships*).
  - Before you enhance infotypes using SE16, check whether the MAINT field is filled in table T777I. If this is the case, the infotype cannot be directly maintained and it is not possible to add an enhancement.



**At the conclusion of this topic, you will be able to:**

- **Create a customer-specific infotype for Personnel Planning**

**ABAP Dictionary: Initial Screen**

• Database table	<input type="text"/>
• View	<input type="text"/>
• Datatype	<input type="text" value="HRI9nnn"/>
• Type group	<input type="text"/>
• Domain	<input type="text"/>
• Search help	<input type="text"/>
• Lock object	<input type="text"/>

Display Change Create

1. Create and activate structure

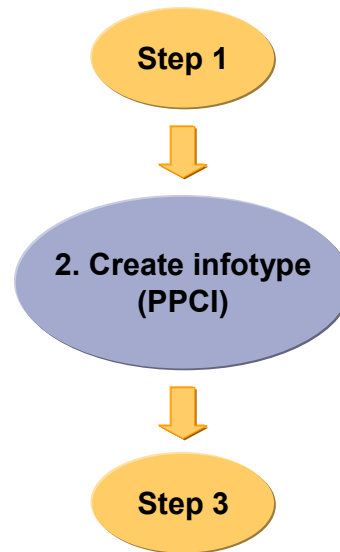
Step 2

Step 3

© SAP AG 2003

## Step 1:

- You must store the infotype-specific fields for your infotype in the system. To do this, create the HRI9nnn structure in the Dictionary.
  - Note that when you do this, you are creating a *Data type* and not a *Database table*.
  - Save and activate the structure.
  - You may need new data elements and domains for your structure.
- If your infotype's data has a repetitive structure, that is, you wish to create a table infotype, you must create the PT9nnn structure for the table part of your infotype. If your table infotype is not composed solely of table entries, but also of individual fields, you must also create the HRI9nnn structure.



© SAP AG 2003

## ■ Step 2:

- Call the Personnel Planning infotype copier (transaction PPCI).
- Enter the four-digit infotype number 9nnn in line with the structure you created and select *Create*.
- When you create the infotype, you can choose between field infotypes and table infotypes. You can also determine whether the infotype is language- and/or country-specific.
- Create a new infotype. When you do this, the infotype copier automatically creates the following objects:
  - Database table HRP9nnn
  - Module pool MP9nnn00
  - Table index
  - 3 screens (1000, 2000, 3000) for the module pool
  - The structure P9nnn
  - The CUA interface- A dialog module (RH\_INFOTYP\_9nnn)
- When you create new infotypes, the following table entries are also automatically maintained:
  - T777D Infotype check type
  - T77ID Enhancements for T777D
  - T778T Infotypes
  - T777T Infotype texts
  - TDCT Dialog modules

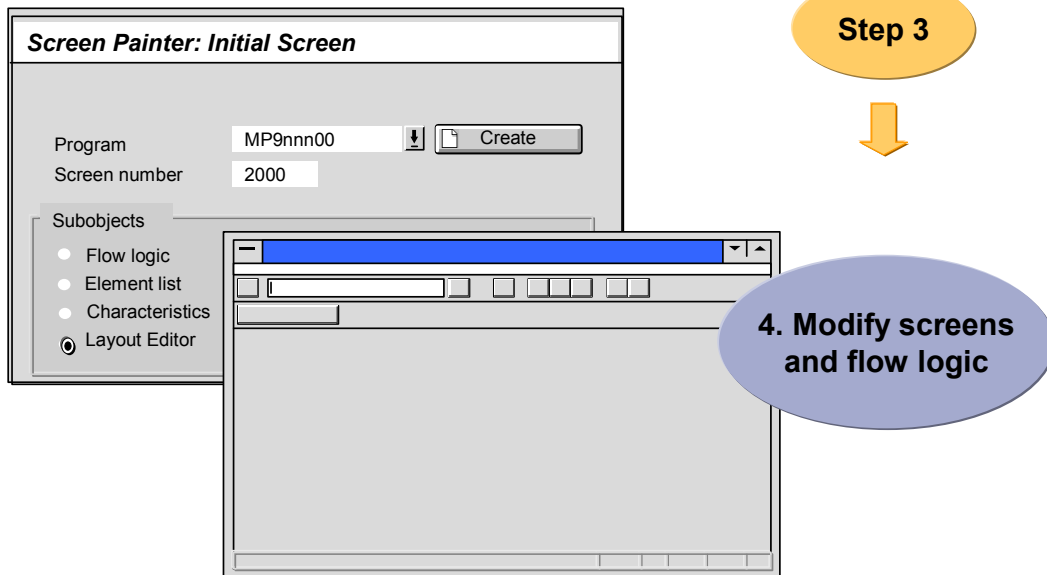
Step 2

3. Maintain tables

Step 4

© SAP AG 2003

- After you create the infotype, entries must be made in the following tables:
  - T777I Infotypes per object type
  - T777Z Time constraint of infotypes
  - T77CD Customer-specific infotype settings
  - T77NI Country-specific infotypes
- Step 3: Select *Check*.
- On the following screen, position the cursor on the table you would like to maintain and select *Change*.
- You can also open these tables in Customizing (path: *Personnel Management* → *Organizational Management* → *Basic Settings* → *Data Model Enhancement* → *Infotype Maintenance*).
- **Note:**  
 You must create the create screen 7000 if you wish to maintain the new infotype using the transaction PPOME. To do this, select *Infotype* → *Create subscreen*. The entry in table T77ID is also enhanced. To make an assignment to a tab, open the IMG (path: *Personnel Management* → *Organizational Management* → *Hierarchy Framework*), and maintain the entries *Adjust Tab Pages in Detail Area* and *Integrate New Infotype* for the scenario OME0.



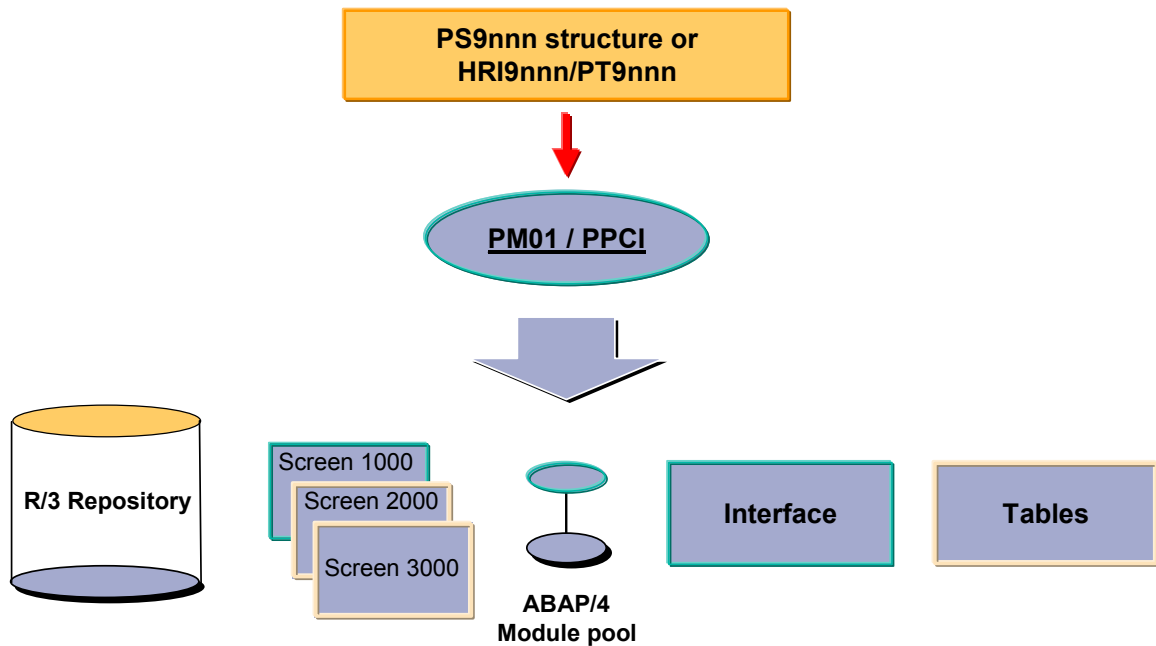
© SAP AG 2003

## ■ Step 4:

- Modify the new screens to suit your needs.
- To do this, edit the layout, the flow logic, and the field characteristics (modification groups).
- You can define infotype-specific check modules in flow logic in include MP9nnn20.
- The default value 036 is set for the first modification group. When you start the report RHGROUP1 using this value, you establish for which functions (for example, create, change, and insert) the corresponding field is ready for input for this value. If you want to use another entry control, use the report RHGROUP0. This calculates the value for the first modification group, based on the function codes you select.

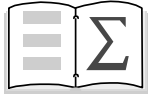


- **Creating New Infotypes - Overview**



© SAP AG 2003

- To sum up: There are three steps for creating customer-specific infotypes:
  - Definition of the info-specific fields in the structures PS9nnn and HRI9nnn in the Dictionary
  - Creation of the central objects of the infotypes using transactions PM01 and PPCI (Personnel Administration and Personnel Planning infotype copiers)
  - Customer-specific modifications and controls



**You are now able to:**

- **Add customer-specific fields to standard infotypes**
- **Create a customer-specific infotype in Personnel Administration**
- **Create a customer-specific infotype in Personnel Planning**



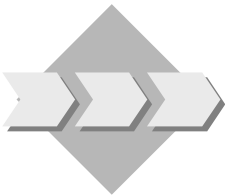
## Unit: Customer-Specific Infotypes

### Topic: Creating an Administration Infotype



At the end of this exercise, you will be able to:

- Create and activate the PS9nnn structure
- Maintain the T582A table
- Modify the infotype single screen



Your company's Personnel Department would like to keep a record of home work centers for individual employees.

- 1-1 Create the *Home work center* (##) infotype for Personnel Administration. Use infotype number **95##**. (## = group number)
  - 1-1-1 Enter the following fields for the infotype:
    - Field 1: **COMP** *Fulltime*, new data element: **ZCOMP##**, domain: CHAR1\_X.  
This field indicates whether the home work center is fulltime.
    - Field 2: **EQUI** *Equipment*, new data element: **ZEQUI##**, domain: CHAR8.  
Use this field to record IDs for possible equipment for the home work center.
  - 1-1-2 Maintain the infotype characteristics (T582A). You can use the characteristics of infotype 0002 as a template.
  - 1-1-3 Convert the *Fulltime* field to a checkbox using the Screen Painter.
  - 1-1-4 Test your infotype using personnel master data maintenance (transaction PA30).
- 1-2 (optional) Enhancement of flow logic of the infotype for the PBO event using a check table for the *Equipment* field.
  - 1-2-1 In your structure **PS95##**, create the foreign key relationship for the *EQUI* field. Enter the **ZZEQUI** table as the check table.
  - 1-2-2 Change the screen for the detail screen for your new infotype, so that the corresponding descriptive text appears next to the *Equipment* field.
  - 1-2-3 Test your development in HR master data maintenance.

- 1-3 (optional) Enhancement of flow logic of the infotype for the PAI event.
  - 1-3-1 Enhance the flow logic of the detail screen for your new infotype so that it is only possible to activate the *Fulltime* field if the *Part-time worker* field is not selected in infotype 0007 *Planned Working Time*. If this field is selected, error message 855 is to be issued from the *RP* message class.
  - 1-3-2 Test your development in personnel master data maintenance.

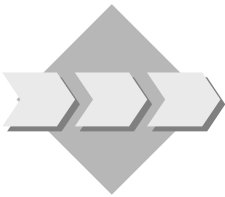


## Unit: Customer-Specific Infotypes

### Topic: Creating a Table Infotype



- Create and activate the required structures for Personnel Planning infotypes.
- Become familiar with transaction PPCI.
- Maintain tables T777I and T777Z.



Data on the size and equipment of offices is required for office planning. Set up the system so it is possible to store this data.

2-1 Create the *Office equipment ##* infotype as a table infotype for Organizational Management. Use infotype number 96##. (## = group number)

2-1-1 Insert the following fields in the non-table part of the infotype:

Field 1: **AREA** *Area*, new data element: ZAREA##, datatype DEC, length 8, 2 decimal places.  
This field stores the area of an office.

Field 2: **NWORK** *Number of work centers*, new data element: ZNWORK##, data type DEC, length 5, no decimal places.  
This field records the maximum number of work centers in an office.

2-1-2 Insert the following fields in the table part of the infotype:

Field 1: **SNWRK** *Work center number*, new data element: ZSNWRK##, data type DEC, length 5, no decimal places.  
This field stores the sequence number of the work centers in an office.

Field 2: **EQUI** *Equipment*, data element: ZEQUI## from the previous exercise.  
This field contains the identifier for the equipment for a work center.

2-1-3 Extend the data model of Organizational Management so that you can maintain your new infotype for the object type **O** *Organizational Unit*.

2-1-4 Test your new infotype in expert mode for the organizational unit ## *Controlling*.





## Unit: Customer-Specific Infotypes

### Topic: Creating an Administration Infotype

#### Note:

- Create all development objects as local objects.
- You must activate the objects before you can test your developments.

1-1 Call transaction **PM01** and enter **95##** in the *Infotype number* field. (## = group number)

1-1-1 Select *PS Structure* and choose *Create*. This automatically takes you to the Data Dictionary structure maintenance screen. In the *Short Description* field, enter the name of your infotype, for example, *Home work center ##*. Enter the field name in the *Components* column, and enter the data element name in the *Component type* column.

Field 1: Name COMP, data element: ZCOMP##

Field 2: Name EQUI, data element: ZEQUI##

#### To create the data element:

Double-click the data element name to go to create mode for component types. Select *Data element* in the dialog box that appears.

Enter relevant short descriptions for your data elements, enter the name of the referenced domain - CHAR1\_X or CHAR8 –, maintain the field labels and the documentation, if relevant.

Save and activate both data elements.

Save and activate the structure.

Select F3 to go back to the *Create Infotype* transaction and select *Create All*.

1-1-2 Maintain the infotype characteristics (T582A) by copying the characteristics from infotype 0002. In the copied entry, correct the name of the infotype and the time constraint. 2 is the correct time constraint for your infotype.

Select F3 to go back to the *Create Infotype* transaction.

1-1-3 To convert the *Fulltime* field to a checkbox: In the *Create Infotype* transaction, select *Screen* under *Subobjects* and enter the number *2000*. Select *Edit*. This takes you to the *Screen Painter*. Select the Graphical layout editor from *Utilities* → *Settings*.

Select *Change* and the *Layout Editor* subobject. Select the field and the field text of the COMP field, and choose **Edit → Convert → Checkbox → Left button**. Save and activate the screen.

1-1-4 To test the new infotype: Call HR data maintenance (transaction PA30), enter an existing personnel number (for example, 1000), enter the number of your infotype (95##), and select *Create*. Enter any data, save, and test the different functions in personnel master data maintenance, such as *Change*, *Copy*, *Delimit*, *Delete*.

1-2 (optional) Enhancement of flow logic of the infotype for the PBO event.

1-2-1 Call transaction **PM01** and enter 95## in the *Infotype number* field (## = group number). Select *PS Structure* and choose *Edit*. This automatically takes you to the Data Dictionary structure maintenance screen. Select the line with the *EQUI* field, choose the *Entry help/check* tab, and select the *Foreign Keys* icon.

The following foreign key relationship is required:

<u>Check table</u>	<u>Check table field</u>	<u>Foreign key table</u>	<u>Foreign key field</u>
ZZEQUI	MANDT	SYST	MANDT
ZZEQUI	EQUI	PS95##	EQUI

1-2-2 Change the screen for the detail screen for your new infotype, so that the corresponding descriptive text appears next to the *Equipment* field.

a) In the include **MP95##10**, insert the following ABAP line:

```
tables zzequit.
```

b) Insert the following statement as the first line of the PBO module **P95##**:

```
select single * from zzequit where equi = p95##-equi
                        and langu = sy-langu
```

c) In the layout of screen 2000, enter the field ZZEQUIT-EQUITXT. To do this, call the *Screen Painter* for the screen, go to the *Layout Editor* and select → **Goto → Secondary Window → Dict/Program Fields**. Enter the table name ZZEQUIT, select *Get from Dictionary*, choose the line with the field EQUITXT and select *Enter* to transfer the field *Without Text*. Now use the mouse to position the field on the screen.



1-3 (optional) Enhancement of flow logic of the infotype for the PAI event.

1-3-1 Enhance the flow logic of the detail screen for your new infotype so that it is only possible to activate the *Fulltime* field if the *Part-time worker* field is not selected in infotype 0007 *Planned Working Time*.

- a) In the flow logic for screen 2000 for the PAI event after the PRE\_INPUT\_CHECKS module, enter the following line:

```
*-----*
*   Entry checks                               *
*-----*
*   Insert check modules here:
field p9500-comp on input module p95##_checks.
```

- b) Double-click p95##\_checks to create this new PAI module. Create it using the following ABAP lines in include **MP95##30**:

```
MODULE P95##_checks INPUT.
  SELECT SINGLE * FROM pa0007
    where pernr EQ p95##-pernr
    and   begda le p95##-endda
    and   endda ge p95##-begda.
  IF NOT pa0007-teilk IS INITIAL.
    MESSAGE e855(rp) .
  ENDIF.
ENDMODULE.
```



## Unit: Customer-Specific Infotypes

### Topic: Creating a Table Infotype

2-1 Create the *Office equipment (##)* infotype as a table infotype for Organizational Management. Use infotype number 96##. (## = group number)

2-1-1 To create the structure for the non-table part of the infotype:  
Call the *Data Dictionary*, and create a new structure called **HRI96##**. Use the following field and data element name:

AREA	Data element ZAREA## Data type DEC, length 8, 2 decimal places
NWORK	Data element ZNWORK## Data type DEC, length 5, no decimal places

To create the data element:

Double-click the data element name to go to create mode for component types.  
Select *Data element* in the dialog box that appears.

Enter relevant short descriptions and field labels for your data elements. Select *Built-in type* for the data type information and use the information provided above.

Save and activate both data elements.

Save and activate the structure.

2-1-2 To create the structure for the table part of the infotype:  
Call the *Data Dictionary*, and create a structure called **PT96##**. Use the following field and data element name:

SNWRK	Data element ZSNWRK## Data type DEC, length 5, no decimal places
EQUI	You created the data element ZEQUI## in the exercise on creating an administration infotype.

Create the data element ZSNWRK as described in 1-1-1.

Save and activate the new data element.

Save and activate the structure.

Call transaction PPCI, enter the infotype number 96## and the description "Office equipment ##". Select *Create*. On the next screen, select the *Table infotype* option and then *Create* in the *Infotype* type.

- 2-1-3 Carry out enhancements in the Organizational Management data model so that you can maintain your new infotype for the *Organizational unit* object type **O**:

In transaction PPCI, select *Check*. In the dialog box that appears, position the cursor on line *T777I: ...* in the *Table Entries* section and select *Change*. Go to table maintenance, and position the cursor on infotype 96##. Select the entry and choose *Time constraint* in the dialog structure. Create a new entry for *Time constraint* with the values:

Object type: O  
Infotype: 96##  
Subtype:  
Time constraint: 2

Maintain a new entry for *Infotypes per object type* with the values:

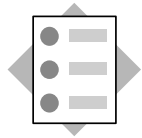
Object type: O  
Infotype: 96##

- 2-1-4 Test the new infotype in expert mode in Organizational Management. Call transaction PP01, or in the SAP Menu, select → *Personnel* → *Organizational Management* → *Expert Mode* → *General*. Select the *Current Plan* plan version and the *Organizational Unit* object type. Call the possible entries help for the *Object ID* field and select the *Other search help* icon. Select the search term search help and enter ## **Cont**. In the infotype table, enter your infotype "Office equipment ##", and create an infotype record. Test other functions of this transaction.



### **Contents:**

- **Calling Routines Using Dynamic Actions**
- **Selection Reports for Fast Entry**
- **Calling Subroutines for Specific Periods**
- **Calling Features From Reports**



**At the conclusion of this topic, you will be able to:**

- **Call subroutines using dynamic actions**
- **Use selection reports for fast entry**
- **Call subroutines for specific periods**
- **Call features from reports**

Course Overview

Data Structures in  
Personnel Administration

The Logical Databases  
PNP/PNPCE

Join and Projection

Specific Commands

Repetitive Structures and  
List Display

Cluster Database Tables in  
HR

Data Structures in  
Personnel Planning

The Logical Database PCH

Customer-Specific Infotypes



**HR ABAP Features**

## Table of Dynamic Actions

ITy	STy	Field	FC	No	S	Variable	function part
0008	__	SPRPS	06	0	_	*-----	Lock indicator set?-----*
0008	__	SPRPS	06	1	P	P0008-SPRPS<>SPACE	
0008	__	SPRPS	06	2	F	COMPUTE_DATE(ZPFORM01)	
0008	__	SPRPS	06	3	I	INS,0019,10	
0008	__	SPRPS	06	4	W	P0019-VTRMN=RP50D-DATE1	

```
REPORT zpform01.
TABLES: RP50D,      "Return fields for dyn. actions
        PRELP.      "HR Master Data Buffer . .
* Calculate date
FORM compute_date.
    RP50D-DATE1 = PRELP-BEGDA -14.
ENDFORM.
```

© SAP AG 1999

- The 'Dynamic Actions' table (T588Z) is a control table that triggers steps taken when an infotype record is maintained. Such steps include maintaining another infotype record, performing a routine, or sending a mail.
- **OP** stands for a user operation where 02 stands for Change, 04 for Insert, and 08 for Delete a record. The values can be added, for example, 06 means that an action is performed if the current record has been changed or inserted.
- **A** stands for an action performed by the system, for example, **P** = Check a condition, **F** = Call a routine, **I** = Maintain an infotype record, **W** = Set default values when inserting a record.
- You can call internal (defined in the module pool) and external routines. In an external routine, the name of the program is given in parentheses after the name of the routine. You cannot enter the USING parameter. The fields of the structure that is not used in the standard system (**RP50D**) are available to return the values from the routine. They can be filled by the routine and then used for default values (W statements). This structure can be enhanced with customer fields in a customer include.
- In this example, the routine COMPUTE\_DATE calculates a date in the subroutine pool ZPFORM01 and places it in field RP50D-DATE1. This date is calculated by subtracting 14 days from the start date of the locked record.



```
REPORT RPLFST00 USING DATABASE PNP.
TABLES PERNR.
INFOTYPES: 0000, 0001, 0002.

DATA: BEGIN OF pernrtab OCCURS 20,
      PERNR LIKE PERNR-PERNR,
      ENAME LIKE P0001-ENAME,
      END OF pernrtab.
. . .

GET PERNR.
. . .

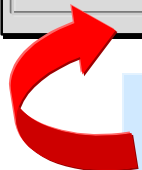
END-OF-SELECTION.
  EXPORT pernrtab TO MEMORY ID 'PERNRTAB'.
```

© SAP AG 1999

- If you use the fast entry function for master data and time data, you can use reports to effect an initial selection of personnel numbers.
- Reports that can be used for fast entry are contained in the 'Selection Reports for Fast Data Entry' table (T588R). If you intend to use customer reports for fast entry, make sure they have been entered in this table.
- The selected personnel numbers are transferred from the report to fast entry using the ABAP memory. The selection report uses the EXPORT statement to store the data as a cluster in the ABAP memory. The calling transaction then imports the data from the ABAP memory.

## Table of HR subroutines

Symb.name	Start date	End date	Module pool	Module	Type
<b>XEDT99</b>	01.01.1991	<b>31.12.9999</b>	RPCEDTX0	EXTCALL	R



```

. . .
SELECT * FROM T596F
  WHERE SNAME = 'XEDT99'
  AND   ENDDA GE PN-BEGDA
  AND   BEGDA LE PN-BEGDA.
ENDSELECT.
. . .
* Dynamic Perform
  PERFORM (T596F-MODNA) IN PROGRAM (T596F-PGMNA)
  USING . . .

```

© SAP AG 1999

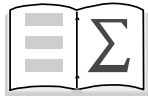
- The HR Subroutines table (T596F) enables you to access various subroutines for specific periods in order to solve a task defined by a symbolic name.
- In the standard system, the symbolic names of subroutines all start with the appropriate country indicator.
- The symbolic name is a freely defined name (such as 'XEDT99') that is used to find a table entry. If a table entry is customer-specific, the symbolic name must start with a special character (for example, &EDT99).
- You first read the table entry valid for a particular period. This contains the name of a program in field **T596F-PGMNA** and the name of a subroutine contained in this program in field **T596F-MODNA**. You then use a dynamic action to call this subroutine from your program.

```

TABLES: PME04.           "Field string for feature ABKRS
DATA:   ret_value(2).    "Return value for feature
* Fill decision fields for feature with values.
PME04-PERSK = P0001-PERSK,
. . .
* Call feature
CALL FUNCTION 'HR_FEATURE_BACKFIELD'
  EXPORTING
    FEATURE           = 'ABKRS'
    STRUC_CONTENT     = PME04
  IMPORTING
    BACK              = ret_value
  EXCEPTIONS
    ERROR_OPERATION   = 2
    NO_BACKVALUE      = 3
    FEATURE_NOT_GENERATED = 4.
    
```

© SAP AG 1999

- This function module reads the decision tree for a feature with the accompanying field contents and determines the return values for the field contents.
- To identify which fields in the field string for the feature (PMENN) are used for the decisions in the feature, view the structure and decision tree in feature maintenance (transaction PE03).
- The field string PMENN must be declared in the data declaration part of your program using a **TABLES** statement. Define a field to contain the return value of the feature.
- In your program, enter values in the decision fields used in your feature.
- Call the function module and enter the name of the feature to be used and the name of the field string. The return value for the feature is transferred to the main program with the **BACK** parameter.
- Possible errors when processing the decision tree:
  - a) An error occurred in the feature (ERROR\_OPERATION)
  - b) No return value is available for the current contents of the decision field (NO\_BACKVALUE)
  - c) The feature was not generated (FEATURE\_NOT\_GENERATED)
- If the return value for a feature consists of a table, use the function module **HR\_FEATURE\_BACKTABLE**.



- You can use a dynamic action to call a subroutine
- You can include your own selection reports for fast entry in the system
- Subroutines can be accessed for specific periods
- Features can also be used in reports



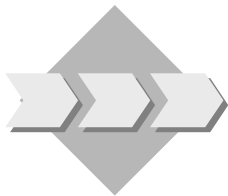
## Unit: HR ABAP Features

### Topic: Creating a Dynamic Action



At the end of this exercise, you will be able to:

- Use a dynamic action to link the maintenance of two infotypes.



If an employee from your company takes on a new function, he or she usually has a six-week introductory period. The Personnel Department would like the system to be set up so that the maintenance screen for the *Monitoring of Tasks* infotype (0019) is also displayed when the *Corporate Function* infotype (0034) is created.

- 1-1 Use the Implementation Guide (IMG) to create a dynamic action for infotype 0034 subtype 9xx (xx = group number). This action is only to be carried out when the administrator creates a new record for infotype 0034.

The new record *Monitoring of Tasks* of infotype 0019 is to be displayed with date type 06 *Introductory period* and a date. Determine the date in a subroutine pool with the formula *Start date for infotype 0034 + 42 days*.

- 1-2 Create a record for employee 50991xx for infotype 0034, subtype 9xx (xx = group number). The validity period is the current year. Test the correctness of your table entries.





## Unit: HR ABAP Features

### Topic: Creating a Dynamic Action

1-1 To create a dynamic action:

*IMG → Personnel Management → Personnel Administration → Customizing Procedures  
→ Dynamic Actions → New Entries*

Make the following entries in the table:

<u>Infotype</u>	<u>Type</u>	<u>Field</u>	<u>FC</u>	<u>No.</u>	<u>A</u>	<u>Variable Function Part</u>
0034	9xx		04	0		*----- Corporate function -----*
0034	9xx		04	1	F	COMPUTE_DATE( <i>program name</i> )
0034	9xx		04	2	I	INS,0019,06
0034	9xx		04	3	W	P0019-TERMN=RP50D-DATE1

Save your entries.

Create a program with the type *Subroutine pool*.

PROGRAM *program name*.

\*-- Declaration

TABLES: RP50D, PRELP.

FORM COMPUTE\_DATE.

RP50D-DATE1 = PRELP-BEGDA + 42.

ENDFORM.

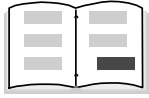
1-2 To test the dynamic action:

*Human Resources → Personnel Management → Administration → HR Master Data  
→ Display*

Choose the *Planning Data* menu and create a new infotype record.







- **This section contains additional material to be used for reference purposes.**
- **This material is not part of the standard course.**
- **Therefore, it may not be covered during the course presentation.**

## **Contents:**

- **Report categories**
- **Exporting clusters**
- **Processing payroll results in Release 4.0**
- **Data Structures in Personnel Planning: Exercises**

© SAP AG 2003

```
TABLES: PCLn.                "Import/export table
INCLUDE: RPCnxxxy0.          "Cluster definition

* Fill cluster-KEY

xy-KEY-FIELD = <VALUE>.
. . . .
* Fill data object

* Export record

EXPORT TABLE1 TO DATABASE PCLn(xy) ID xy-KEY.

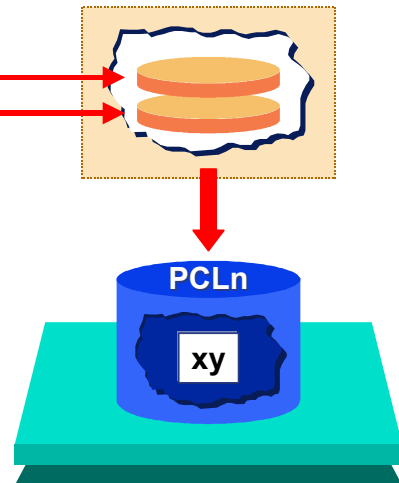
IF SY-SUBRC EQ 0.
  WRITE: / 'Update successful'.
ENDIF.
```

© SAP AG 1999

- The cluster definition is included using the INCLUDE statement.
- The EXPORT command writes one or more data objects with xy-KEY to cluster xy.
- If the export is successful, the return code is 0.

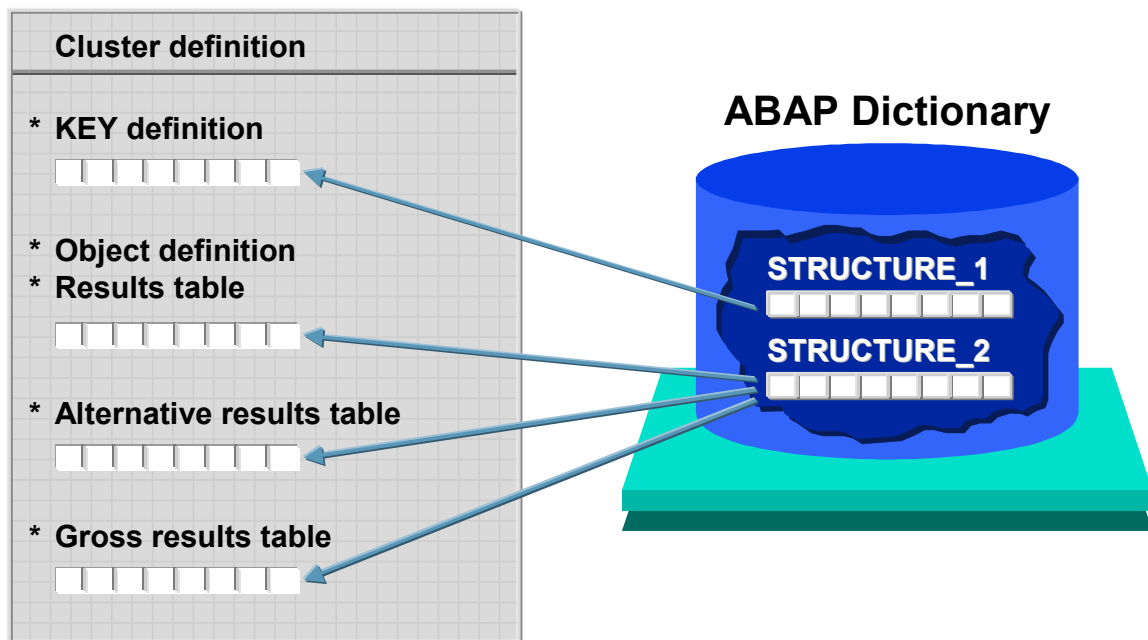
```
* Buffer definition
INCLUDE: RPPXD00.
DATA: BEGIN OF COMMON PART buffer.
      INCLUDE RPPXD10.
DATA: END OF COMMON PART buffer.
. . .
* Data export to buffer
RP-EXP-Cn-xy.
RP-EXP-Cn-xy.
. . .
* Save
PERFORM PREPARE_UPDATE USING 'V'.
. . .
* Buffer Administration Routines
INCLUDE: RPPXM00.
```

**Main memory buffer**



© SAP AG 1999

- If data is exported using macros, the data records are not written directly to the database. Instead, they are written to a main memory buffer. The data is diverted by a USING parameter in the EXPORT statement, which accesses a buffer administration routine.
- If data is exported successfully, the RP-IMP-xy-SUBRC = 0 return code is set.
- At the end of the program, the buffered records must be saved on a PCLn database.
- To save the data, the PREPARE\_UPDATE routine is accessed with USING parameter V.



© SAP AG 1999

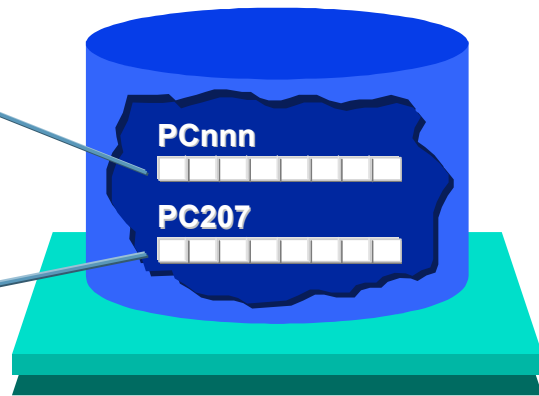
- The cluster definition of payroll results complies with the valid naming conventions. They are stored in the `RPC2xyz0 INCLUDE` reports, where `z` stands for the HR country indicator.
- The cluster key is contained in the `RX-KEY` structure.
- All of the fields in the key and data objects are defined in ABAP Dictionary structures.

```

Cluster definition
REPORT RPC2Rn00.
* KEY definition
DATA: BEGIN OF RX-KEY.
      INCLUDE STRUCTURE PCnnn.
DATA: END OF RX-KEY.

* Object definition
* Results table
DATA: BEGIN OF RT OCCURS 150
      INCLUDE STRUCTURE PC207.
DATA: END OF RT.
    
```

## ABAP Dictionary



© SAP AG 1999

- The fields for the key and objects are defined in ABAP Dictionary structures. This means that you can use them time and time again without jeopardizing consistency.
- Structures used for the cluster definition comply with the PCnnn naming convention, where nnn stands for the alphanumeric characters of your choice.
- The PC200 structure contains the key definition of the results cluster. It consists of two fields, PERNR (personnel number) and SEQNO (sequential number).
- The data definitions of international payroll results are stored in includes RPC2RX00 and RPC2RXX0.

```

DATA: number LIKE PC261-SEQNR.
. . .
CALL FUNCTION 'CD_READ_LAST'
  EXPORTING
    BEGIN_DATE      = PN-BEGDA
    END_DATE        = PN-ENDDA
  IMPORTING
    OUT_SEQNR       = number
  TABLES
    RGDIR           = RGDIR
  EXCEPTIONS
    NO_RECORD_FOUND = 1
    OTHERS          = 2.
. . .
RX-KEY-PERNR = PERNR-PERNR.
RX-KEY-SEQNR = number.
RP-IMP-C2-RX.
. . .
LOOP AT RT.
  WRITE: / RT-LGART...
ENDLOOP.

```

© SAP AG 1999

- This function module determines the current payroll result for a for-period to be evaluated. To determine the correct start date and end date of the for-period, you specify the period by entering the payroll period in the selection screen. If you specify report class XXM00004 in the attributes of your report, the payroll period is entered and the start date (PN-BEGDA) and the end date (PN-ENDDA) are determined using the Payroll Periods table (T549Q).
- You enter the start and end date of the for-period for the evaluation as well as table RGDIR. The function module then gives you the sequential number (OUT\_SEQNR) for the current (A) result of the for-period.
- You then transfer the sequential number to RX-KEY and import the required payroll result with the corresponding macro RP-IMP-C2-xy (xy = cluster ID).
- You process the wage type tables for the payroll result sequentially using LOOP.
- You can also use the following function modules:
  - CD\_READ\_PREVIOUS  
(reads the record that precedes the payroll record)
  - CD\_READ\_PREVIOUS\_ORIGINAL  
(reads the last original result that precedes the original payroll result)

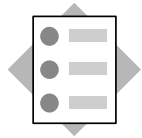
```

*Table containing evaluation periods
DATA: BEGIN OF EVPDIR OCCURS 100.
      INCLUDE STRUCTURE PC261.
DATA: END OF EVPDIR.
...
CALL FUNCTION 'CD_EVALUATION_PERIODS'
  EXPORTING
    BONUS_DATE      = <Bonus date>
    INPER_MODIF     = <Period par.>
    INPER           = <In-period>
    PAY_TYPE        = <Payroll type>
    PAY_IDENT       = <Payroll ID>
  TABLES
    RGDIR           = RGDIR
    EVPDIR          = EVPDIR
  EXCEPTIONS
    NO_RECORD_FOUND = 1
    OTHERS          = 2.
    
```

© SAP AG 1999

- This function module fills table EVPDIR (evaluation periods) with payroll result data from the selected period. Descriptive data from the current (A) and previous (P) result is written to table EVPDIR.
- **Period parameter:** Specifies the period in which the payroll runs for a payroll area. Just one period parameter is assigned to a payroll area. Payroll periods are defined for each period parameter.
- **In-period:** The period in which a payroll result is created.
- Payroll type: Indicator for "Type of Payroll Result" (according to table T52BX)
  - " " = Regular payroll result (payroll result for a payroll period)
  - "A" = Bonus accounting (bonus date transferred)
  - "B" = Correction run
  - "C" = Manual check
- **Payroll identifier:** Indicator used to make a distinction between various special accounting runs performed on the same day
- You can also use the following function modules:
  - CD\_READ\_PREVIOUS  
(reads the record that precedes the payroll record)
  - CD\_READ\_PREVIOUS\_ORIGINAL  
(reads the last original result that precedes the original payroll result)





**At the conclusion of this topic, you will be able to:**

- **Describe the special features of external object types**
- **Describe the special features of external infotypes**

- **Definition of external object types and relationships**

- **External object types** = Object types that are not stored in Personnel Planning structures (that is, no record in HRP1000), but are only referred to in the context of relationship records (only records in HRP1001)

- 
- **Table T77EO** (IMG path: *Maintain Object Types - External Object Types*)

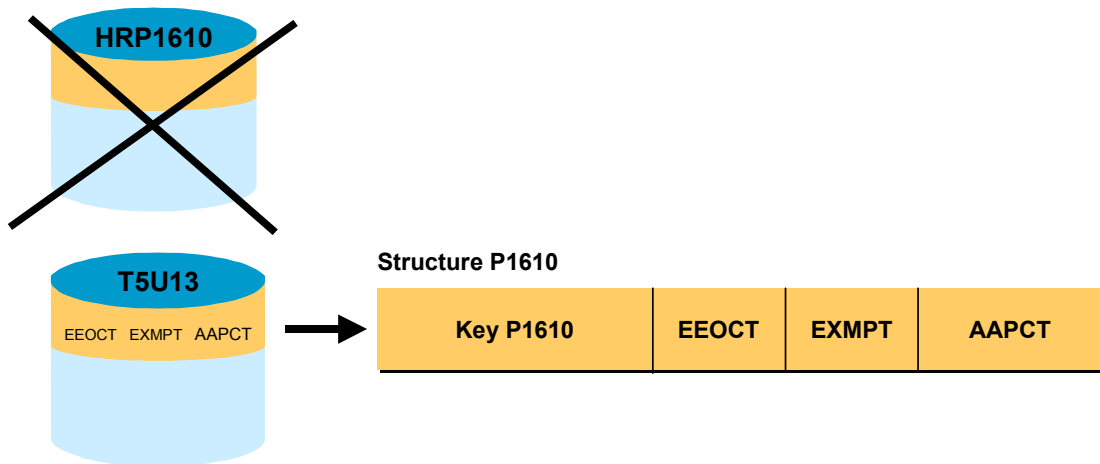
Object type	Key structure	Inversion	Interface
P	PKEYS	X	RHPREL00
K	PKEYK		RHKOST00
...			

Depending on the key structure, a distinction is made between external object types with a physical inverse relationship in HRP1001 (for example, persons from PA) or without (for example, cost centers from CO).

- 
- **External relationships** = Simulation of complete relationships, that is, no relationship direction is physically stored in HRP1001 (**table T77EV**; IMG path: *Maintain Relationships - Allowed Relationships - External Relationships*)

- **Definition of external infotypes**

- ➔ **External infotypes** = Simulation of information on Personnel Planning objects that are only stored in Administration Customizing tables and not as a Personnel Planning infotype in transparent infotype tables HRPnnnn
- ➔ For example: Additional job data for the USA, infotype 1610 and T5U13



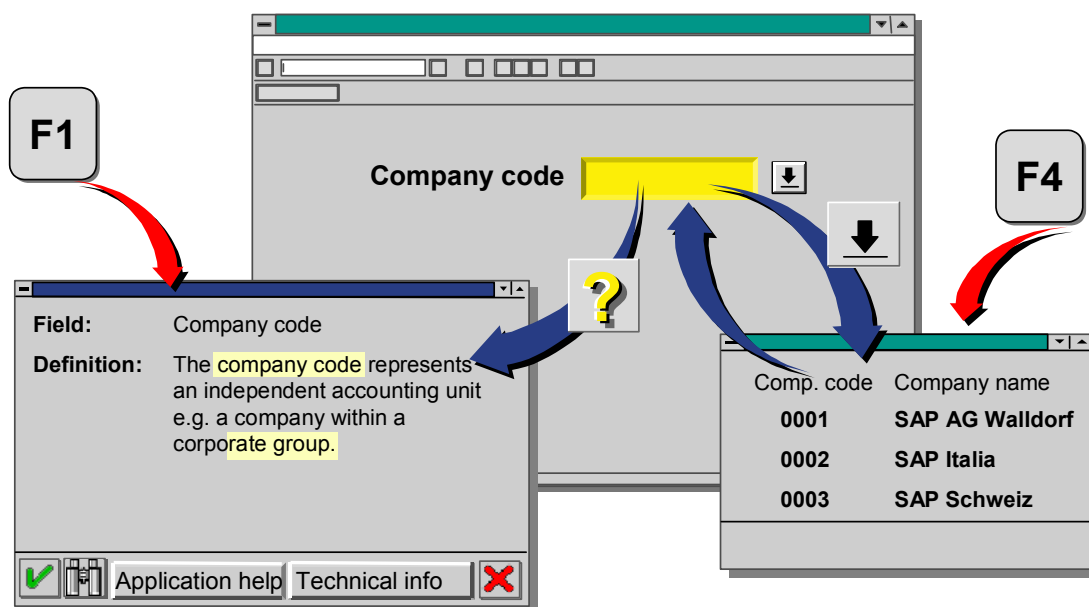
© SAP AG 2003



## **Contents:**

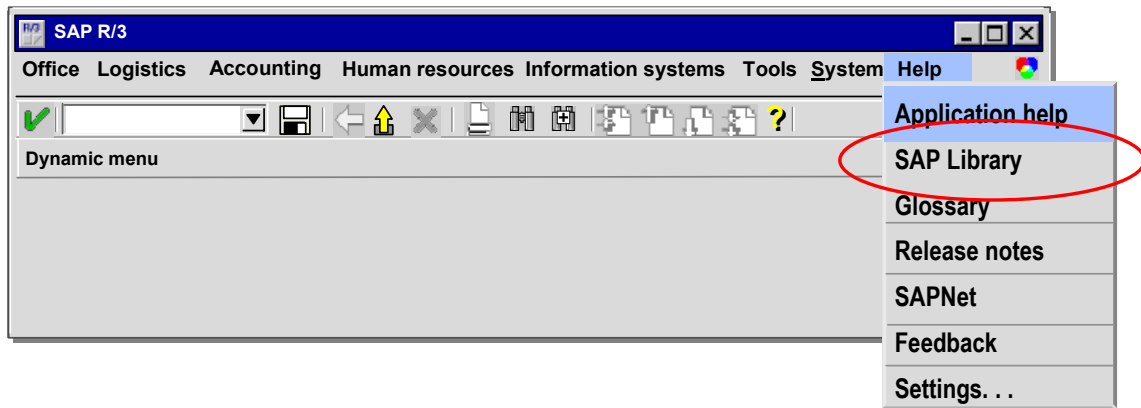
- **Human Resource Process**
- **Enterprise Structure**
- **Personnel Structure**
- **Infotypes**
- **Time Management**
- **Payroll**

© SAP AG 1999



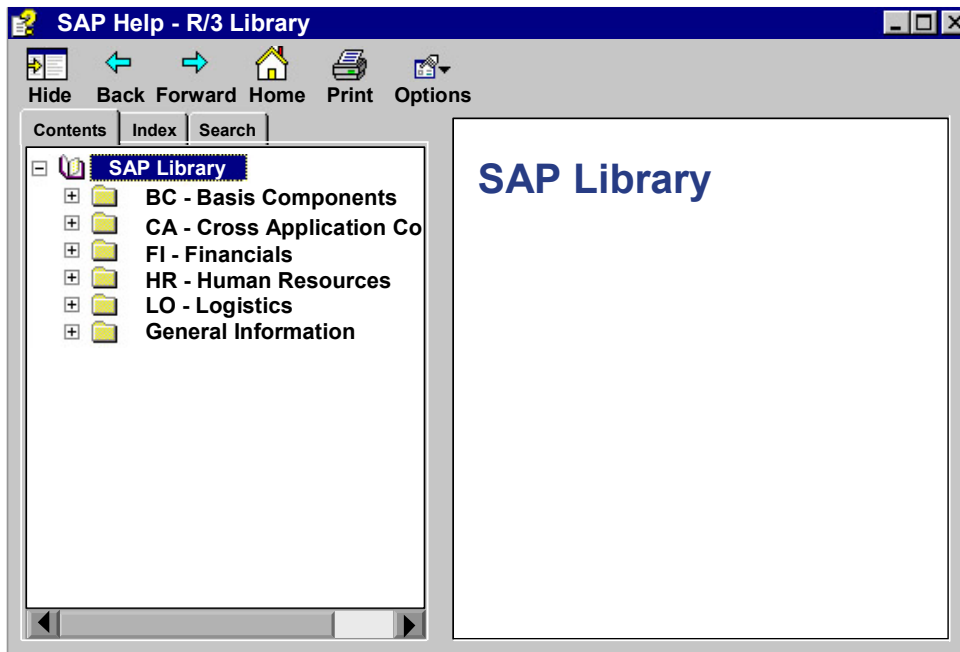
© SAP AG 1999

- For help on a field, place the cursor on the field and press [F1] or choose the help symbol - a question mark.
- You can access glossary information for terms highlighted in the help text.
- To do this, point to the term and click on the highlighted term.
- For further information on possible entries and the field definition, choose *Application Help* or *Technical info*.
- Access a list of possible entry values or search helps to help you search for the data you require, by placing the cursor on the entry field and pressing [F4] or choosing the possible values icon at the end of the field.
- To transfer a value from the list of entry values to the entry field, place the cursor on the value and choose *Choose*, choose [F2], or point and double-click the left mouse button.



© SAP AG 1999

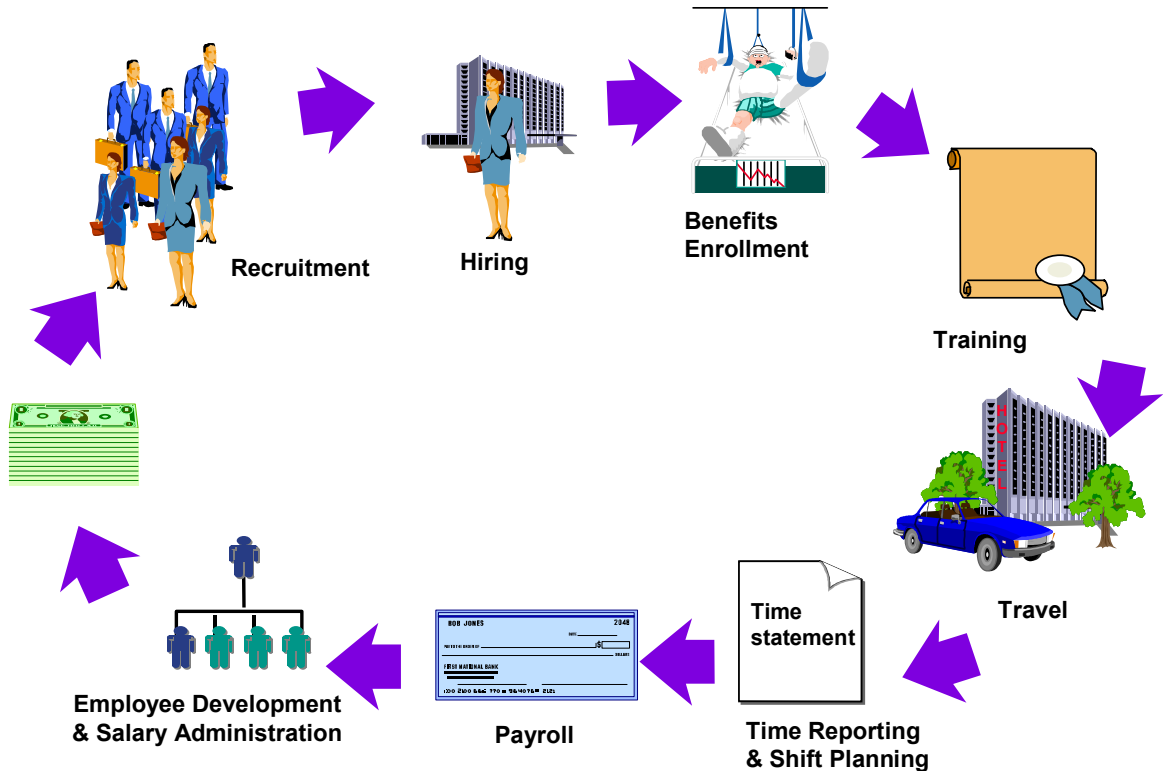
- There are many ways to access help within the SAP System.
- *Getting Started* is a good introduction to the basic concepts of the SAP system, for example, logging on and off, using the SAP System, working with several sessions, and so on.
  - It describes common screens.
  - It describes common actions.
- *Getting Started* is part of the SAP Library.
  - It is in HTML format.
  - It contains links to other help sections.
- *Release Notes* explain what has changed between releases of the SAP System.



© SAP AG 1999

- Most SAP Help files are HTML files.
- *SAP Library* is an online library of the entire R/3 documentation. You can search for the information you need or drill down through applications.
- *Application Help* displays specific information from the SAP Library. The information displayed is determined by which screen you are in.
- The *Glossary* defines technical terms used within the SAP system.

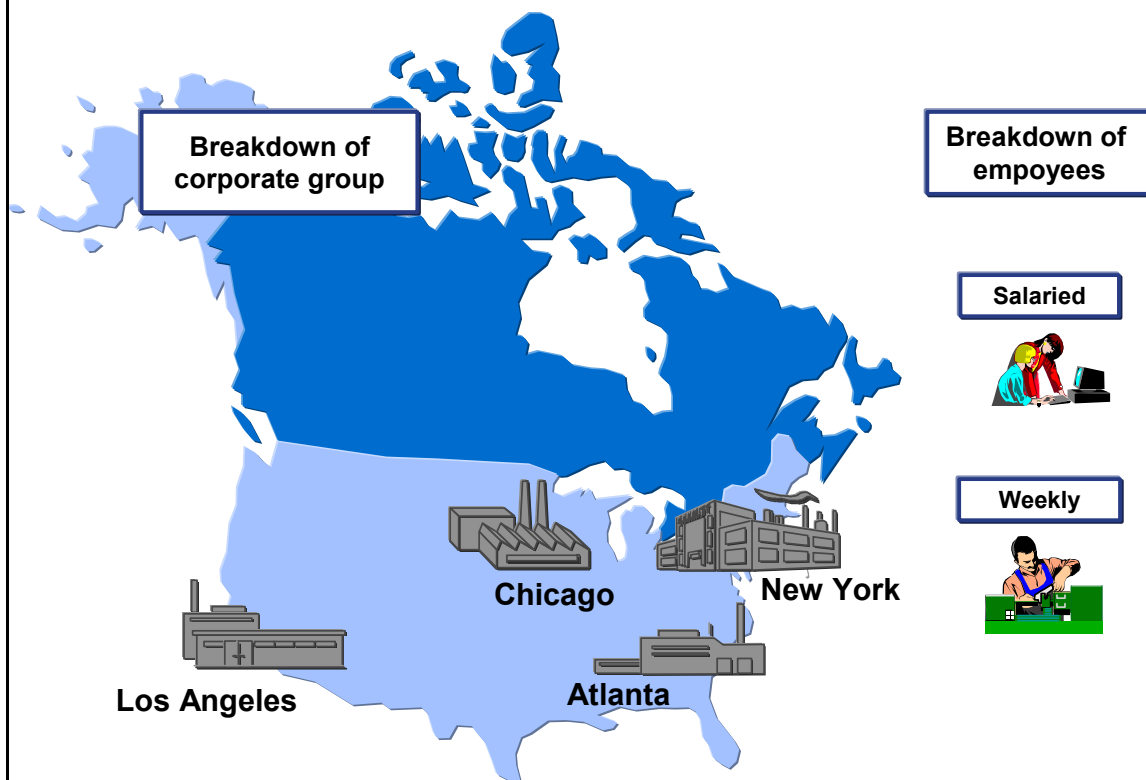




© SAP AG 1999

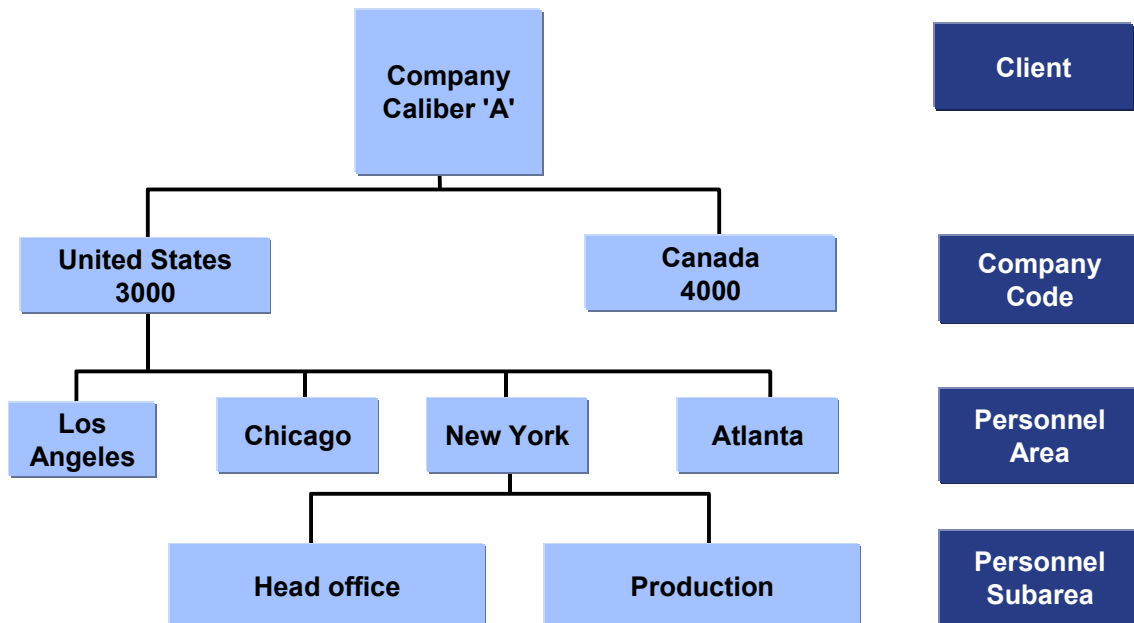
■ Every component of this process is supported by the SAP Human Resources system:

- Recruitment
- Hiring
- Benefits Enrollment
- Training
- Travel Expenses
- Time Evaluation
- Shift Planning
- Payroll
- Employee Development
- Compensation Management
- Cost Planning
- Reporting



© SAP AG 1999

- A personnel management system should enable you to reproduce company hierarchies and the relationships within the hierarchies, and to store and manage employee data.
- The organizational hierarchy refers to the division of a corporate group into company code, personnel areas, and personnel subareas.
- The first step in recording personnel data in the SAP System is to assign the employee within the company structure.
- The employee is assigned to a work area, a specific status (for example, active or inactive) and type (for example, salary or hourly).
- Employees are also assigned a position within the organization - within employee groups and employee subgroups.

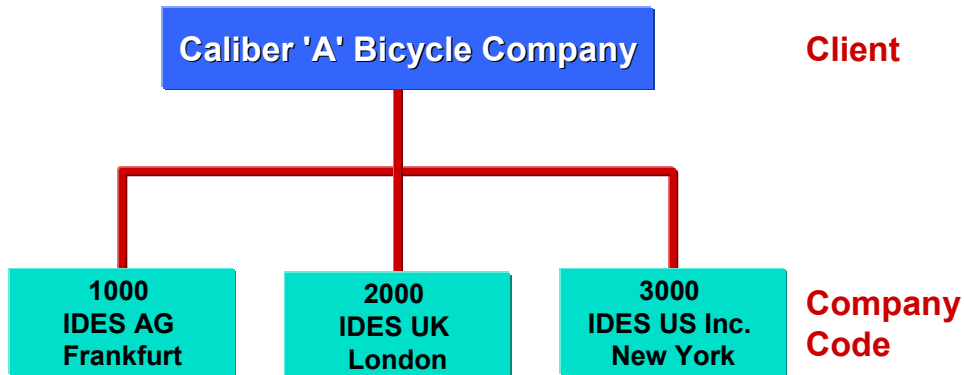


© SAP AG 2003

- The company structure for Human Resources consists of the following:
  - Client
  - Company code
  - Personnel area
  - Personnel subareas
- A **client** can be valid for one or more company codes.
- A **company code** is defined in Accounting. Legally required financial statements, such as balance sheets and profit and loss statements, are created at the company level.
- The **personnel area** is used exclusively in Human Resources and is unique within a client. Each personnel area must be assigned to a company code.
- The final element of the company structure, unique to Human Resources, is the **personnel subarea**.

**The client is an independent legal and organizational unit of the system.**

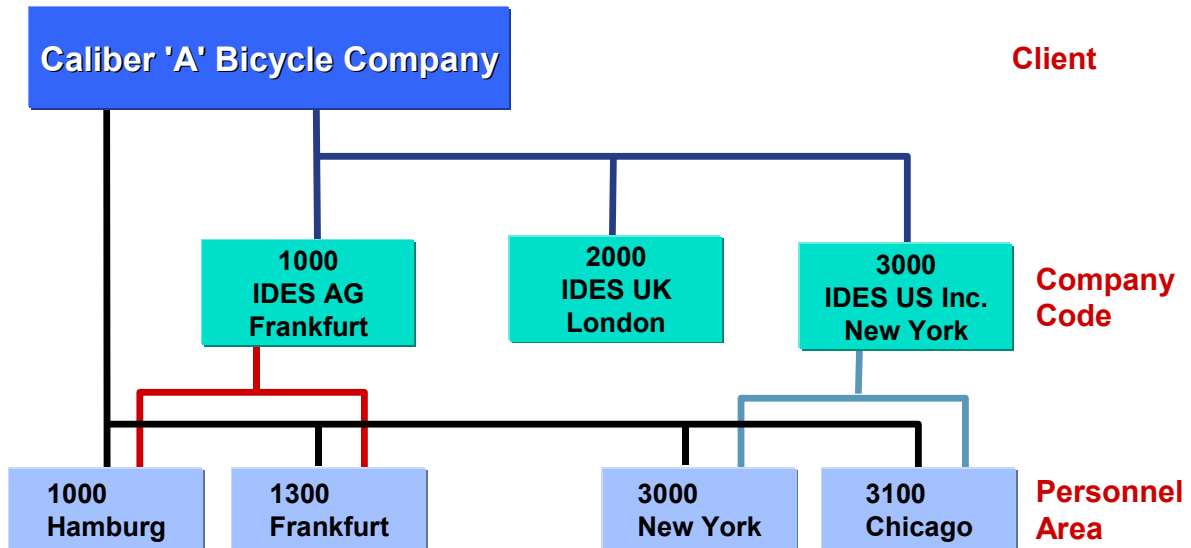
**The company code is an independent company with its own accounting unit - a company which draws up its own balance sheets.**



© SAP AG 1999

- In legal and organizational terms, the client is a self-contained unit within the system. You should consider the following points before you decide whether or not to set up a client:
  - It is not possible to access personnel data in other clients; this protects data from being accessed by unauthorized users.
  - You cannot assign access authorization or run evaluations for all clients.
  - There is no exchange of data between clients.
  - If an employee changes clients, you have to create the personnel number again.
- The company code is a self-contained unit in legal terms, for which you can draw up a complete set of accounts. The company code is the highest level of the company structure. If you also use the Controlling, Financial Accounting, Materials Management or Sales and Distribution components, the company code must be set up in consultation with these applications. This does not impose any restrictions, however, as all important control information for the Human Resources system is defined at personnel subarea level.
- All of the Caliber 'A' Bicycle Company's United States employees belong to company code 3000.

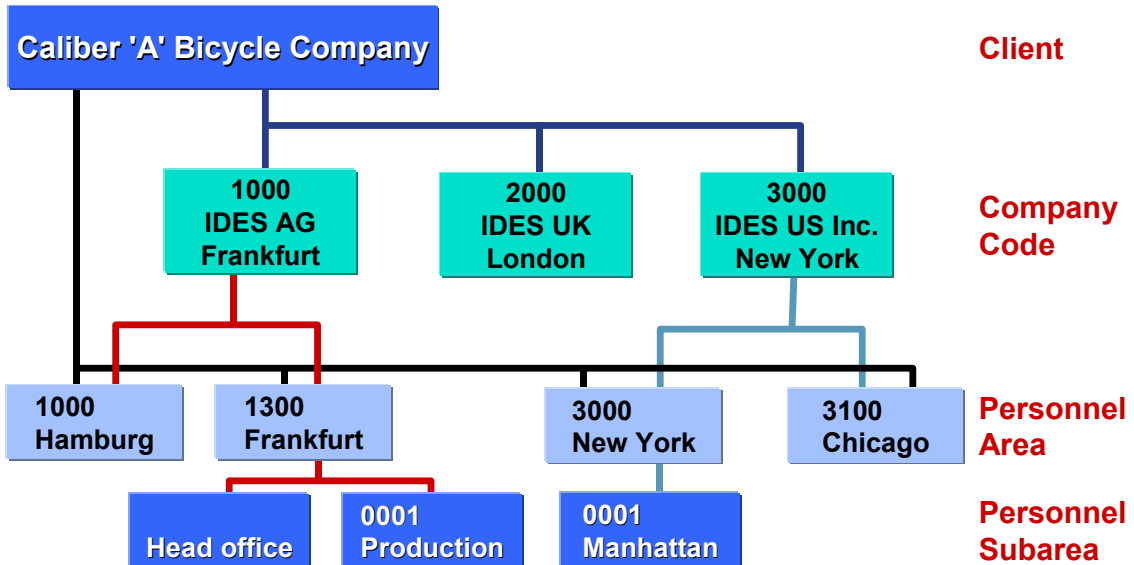
A personnel area is a Personnel Administration-specific unit and is the subunit of the company code.



© SAP AG 1999

- A **personnel area** is a Personnel Administration-specific unit and is the subunit of the company code. The individual personnel areas in a company code have four-digit alphanumeric identifiers.
- The personnel area has the following functions:
  - It allows you to generate default values for data entry, for example, for the payroll area.
  - It is a selection criterion for reporting.
  - It constitutes a unit in authorization checks.

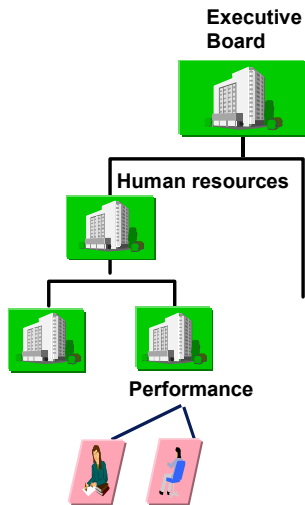
Personnel subareas are subdivisions of personnel areas.  
The organization of the most important subareas of personnel administration takes place at this level.



© SAP AG 1999

- **Personnel subareas** are subdivisions of personnel areas. The organization of the most important subareas of personnel administration takes place at this level. The personnel subarea is assigned a four-character alphanumeric identifier. The control features are stored according to the country.
- The main organizational functions of the personnel subarea are as follows:
  - To specify the country grouping. Master data entry and the setting up and processing of wage types and pay scale groups in payroll depend on the country grouping. The grouping must be unique within a company code.
  - To assign a legal person which differentiates between companies in legal terms.
  - To set groupings for Time Management so that work schedules and substitution, absence and leave types can be set up for individual personnel subareas.
  - To generate a default pay scale type and area for an employees basic pay.
  - To define a public holiday calendar.
  - To define subarea-specific wage types for each personnel area.

## Organizational view



## Administrative view

Organizational Assignment (0001)

PersNo 1000  
Name Anja Muller  
Financial accounting

01.05.1998 - 31.12.9999

Enterprise structure

Pers. structure

Org. structure

Position

Percentage 100

Job 50000064

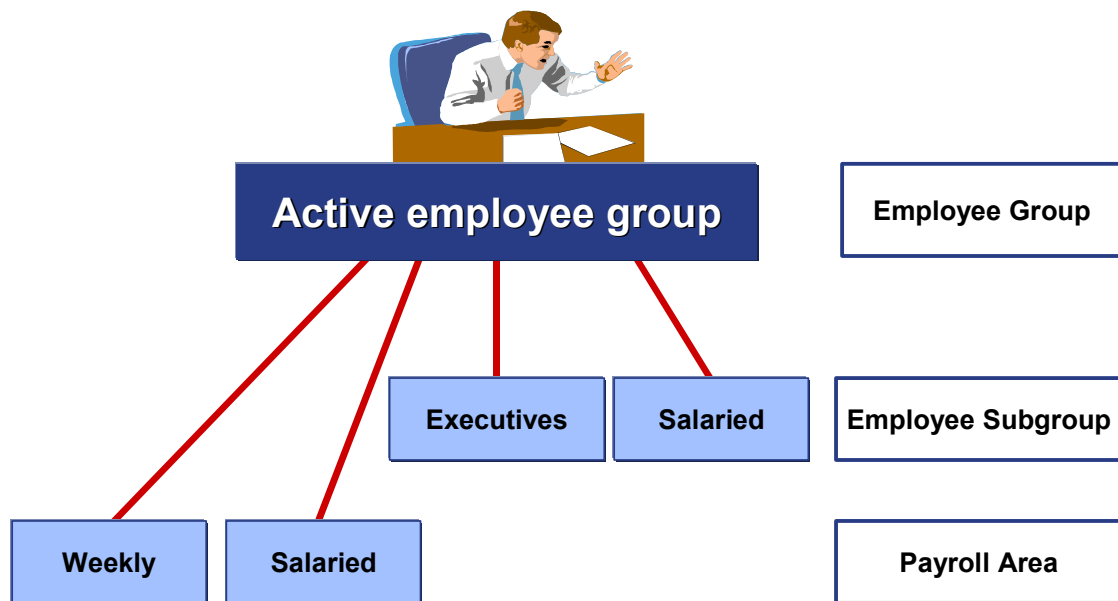
Org.unit 50000827

Org.key 0001

Administrator

© SAP AG 1999

- When an employee is hired, he or she is assigned to an organizational unit, a job and a position in the *Organizational Assignment* infotype (0001).
- Organizational unit, job, and position are all components of the organizational plan and are maintained in the *Organizational Management* component.
- You implement the *Organizational Management* component to create a model of your enterprise in the system.

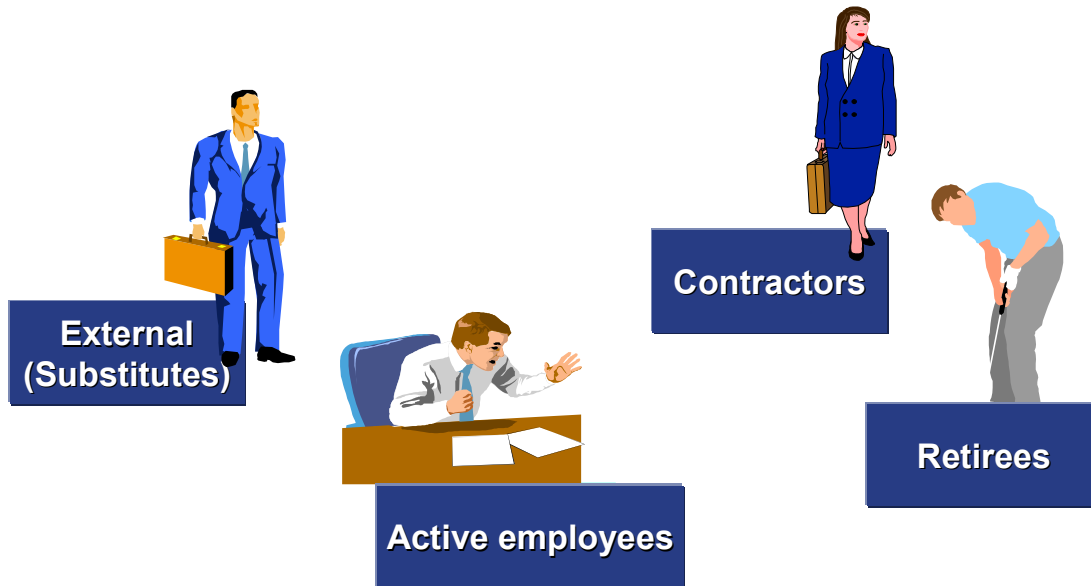


© SAP AG 1999

- The administrative personnel structure is made up of three elements:
  - Employee groups
  - Employee subgroups
  - Payroll areas
- Part of the personnel structure for the Caliber 'A' Bicycle Company is shown above.
  - The active employees belong to the employee group Active.
  - The employee group Active Employees is subdivided into different employee subgroups, for example, salaried employees, trainees, and monthly-wage earners.
  - A payroll area groups together employees for whom payroll is to be run at the same time.
- In the new Customer Service Center of the Caliber 'A' Bicycle Company, all employees will be active. Managers and supervisors will be salaried and customer service representatives will be hourly employees. All employees will receive their checks on a semi-monthly basis.



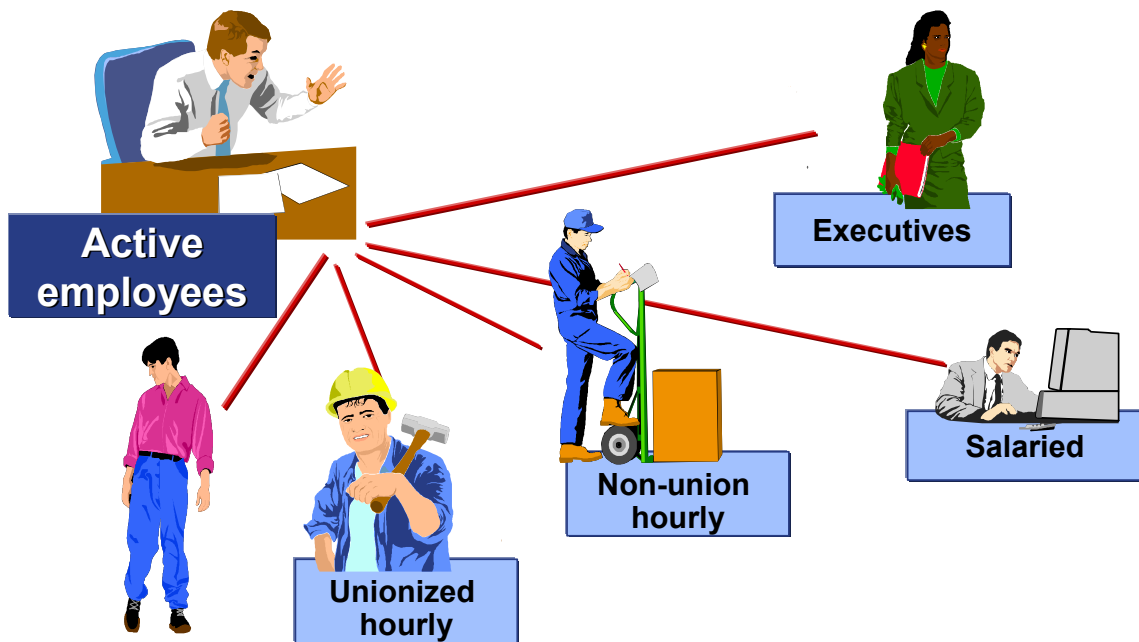
Employees are categorized into various **employee groups**.



© SAP AG 1999

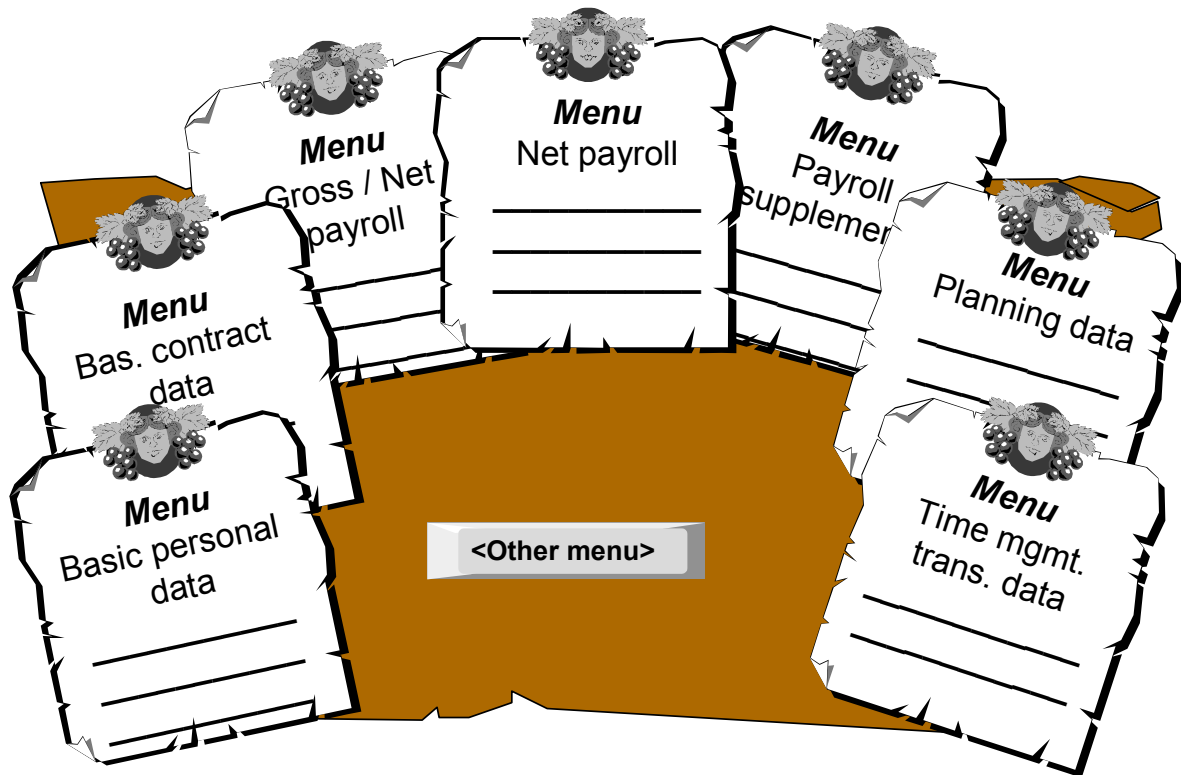
- Employee groups represent different types of employees in an organization.
- The Caliber 'A' Bicycle Company has active employees, retirees, contractors, and external employees.
- Employee groups are used to:
  - Generate default values for payroll area and basic pay
  - As a selection criterion for reports
  - As one unit of the authorization check
- You can use the system-delivered entries to set up employee groups and extend this list to suit particular organizational requirements.

## Employee groups and employee subgroups



© SAP AG 1999

- The employee subgroup is a division of employee groups according to the types of employees.
- The employee subgroup determines:
  - Payroll calculation
  - Work schedule groupings
  - Grouping for personnel calculation rule (Payroll)



© SAP AG 1999

- Infotypes are used in HR to input and store related employee information.
- The most frequently used infotypes are grouped according to function and are contained in menus. One infotype can occur in several menus. Infotypes that are seldom used are not contained in menus, however they can be accessed individually by entering the four-digit infotype number.
- Select the required infotype menu directly using the tab page. If the required menu is not displayed, you can use the forward or backward arrow to search or you can use the pulldown menu to find the required menu.



**Maintain Master Data**

**Personnel number** 1000  
Anja Muller

**Basic personnel data**

☐ Actions  
☐ Personal Data  
☐ Addresses

**Period**

**Direct selection**

**Infotype**  **Type**

 An illustration of a right hand with an orange skin tone, pointing towards the "Period" field of the SAP screen. The hand is wearing a light blue sleeve.

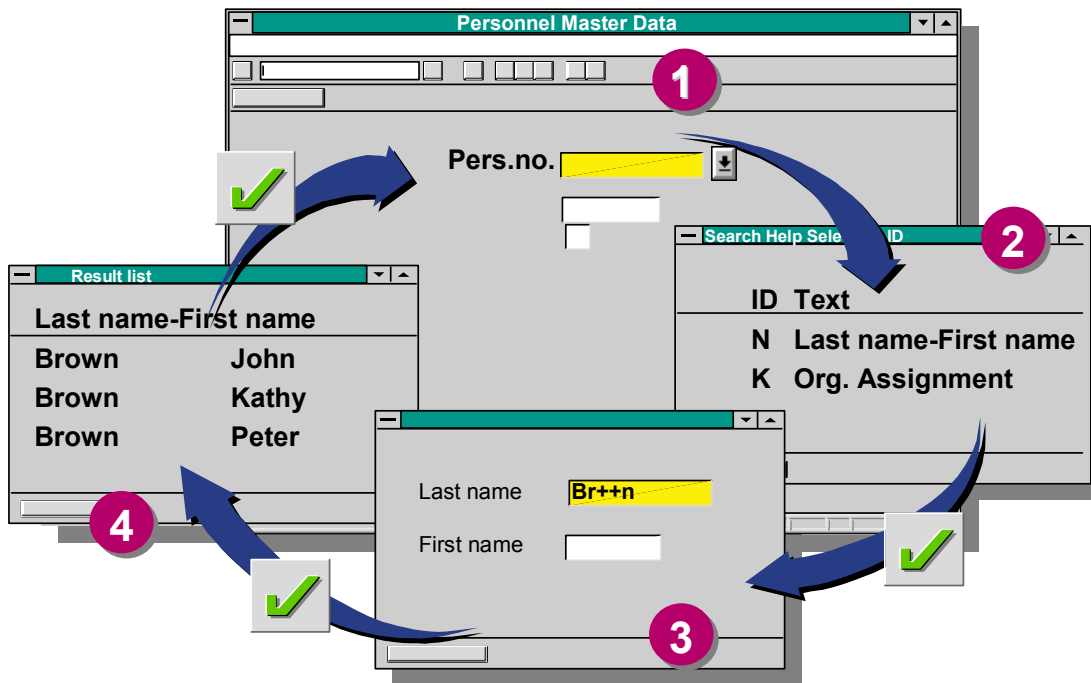
© SAP AG 1999

■ To access a selection screen with an infotype menu:

1. Choose the menu path:

Human resources → Personnel management → Administration → Display master data

2. Enter a personnel number to display information about the specified employee.



© SAP AG 1999

- There are a number of search utilities consisting of various search strings arranged in a particular order. Each search utility has its own identification (ID).
- To search for a personnel number using a matchcode, place the cursor on the personnel number entry field and click on the Possible entries button.
- A list is displayed showing all the available search utilities. You can then choose the required matchcode by double-clicking on it.
- In the subsequent screen you enter search terms, such as last and first name.
- You will then obtain a list of all personnel numbers that meet the search criteria. You can choose the personnel number you require from this list.

**Basic personnel data**

☐ Events

☐ Org. Assignment

☐ ...

☐

**Infotype**

6

**Display**

**Display Addresses (Infotype 0006)**

Pers.no.

Valid

Address type

**Address**

c/o

Street and house no.

City, state and zip

© SAP AG 1999

- If you know the number of the infotype, you can enter it in the Infotype field. Choose the requested processing type and the system will display the selected infotype record.



## **Change (Correct) → No History**

- Correct errors
- Complete missing fields



## **Create → History**

- Create a new record
- Empty input mask
- Fill in entry fields



## **Copy → History**

- Create a new record
- Complete mask from the previous record
- Customize fields
- Specify start date

© SAP AG 1999

- There are three functions to maintain infotypes: change, create, and copy.
- By using the **Change** function, you can correct an existing record without creating a new one. You choose this function if you want to correct the field contents or add additional data. This function does not create a new record, thus no history is available.
- If an employee's personal data has changed, choose the **Create** function to enter new data. A blank input template is displayed. If you create a new record, the old record remains in the system and is delimited if necessary. An infotype history is generated according to the validity periods for the individual records.
- You can also use the **Copy** function to create a new record. The system displays the current record together with the field contents. You can either copy the field contents that has not changed or, if necessary, maintain the data; this means that you do not have to enter the data in a blank template. To continue the infotype history, enter the start date of the new record.
- When you make changes relating to personnel numbers stored in the system, you should always create new records to keep the infotype history up to date. For example, you create a new record when an employee's address changes: the old address is valid until December 31, 1998 - the new address is valid from January 1, 1999.



### **Save**

You must save each infotype record you change or create.



### **Delete**

Allows you to delete entire infotype records.



### **Previous record / Next record**

Allows you to scroll between records within an infotype.



### **List**

Allows you to display a list of all records for an infotype in chronological order.

### **Maintain text**

For information purposes, you can store a text for each infotype record.



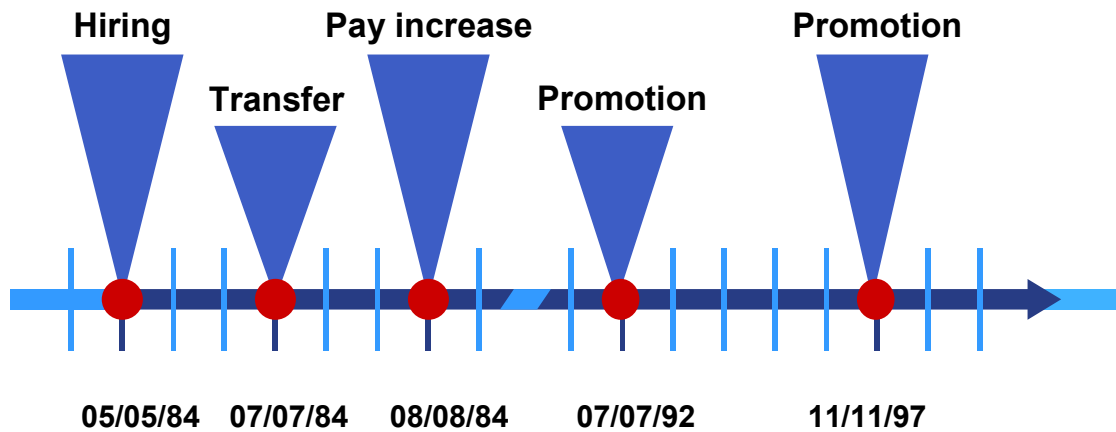
### **Overview**

Also displays infotype records in chronological order.

© SAP AG 1999

- Use the icons when editing infotypes.

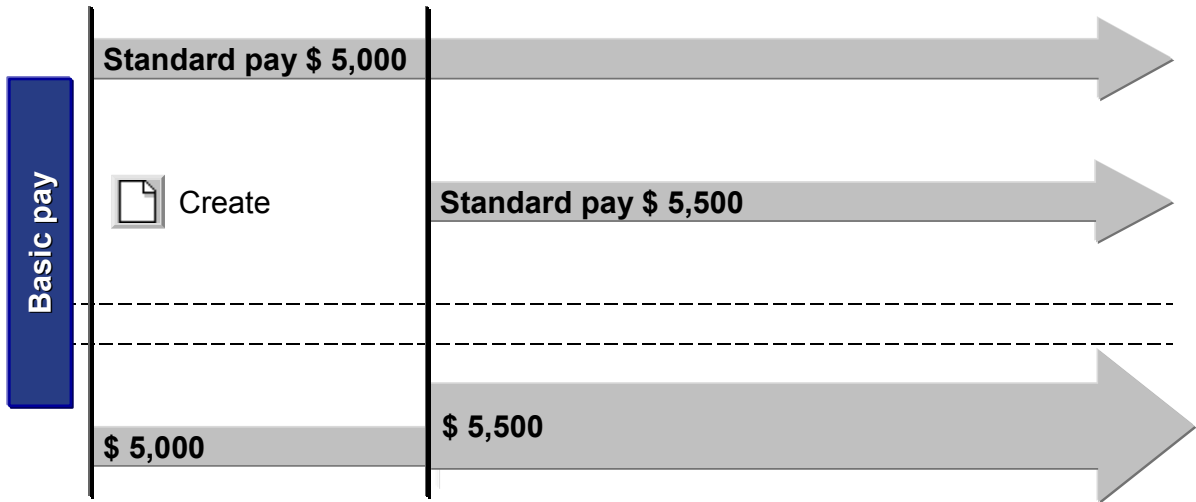




© SAP AG 1999

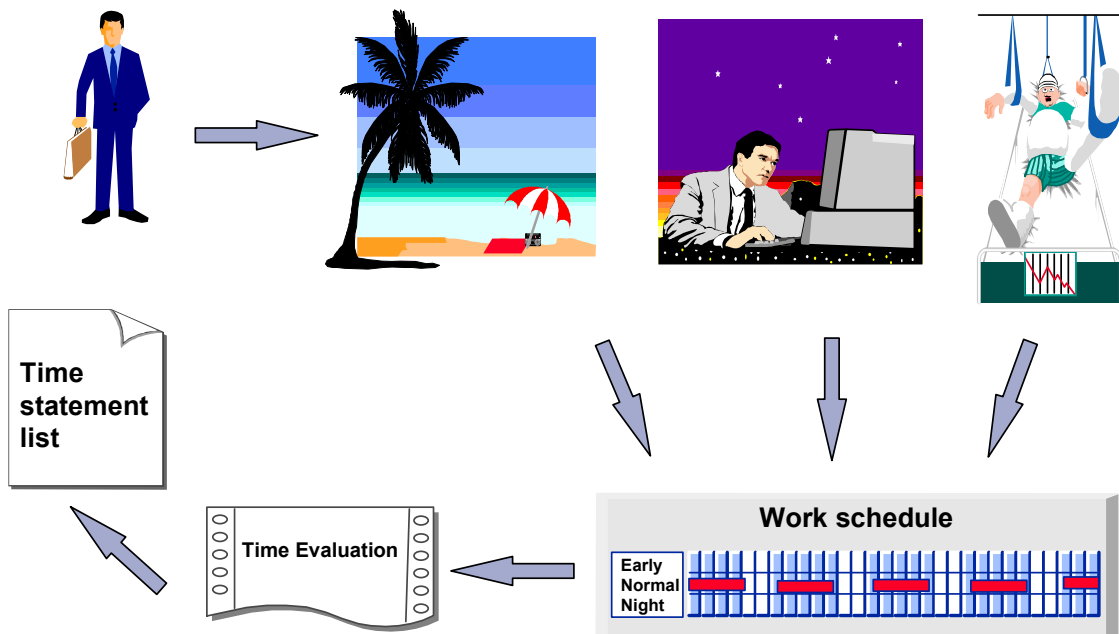
- When you create an infotype, the old data is not lost, but is stored as history for reporting purposes. Entering a specific validity period for each infotype record allows several records to exist at one time for an infotype.

01/01/1996                      07/01/1996                      12/31/9999



© SAP AG 1999

- When you create a new record that must cover the time axes without gaps and overlaps (time constraint 1), the existing record uses the day before the start date of the new record as the validity end date. The new record is then valid until the year 9999.
- If you delete a record that must exist in the system at all times, the previous record is automatically extended.
- You can make a retroactive payroll change and all corrections are applied to the system.



© SAP AG 1999

- Employees are assigned to a work schedule, which specifies the times at which they should be at work.
- There can be exceptions to the work schedule due to vacation or sickness, for example. Exceptions must also be recorded in the system.
- Each employee's working times are evaluated, and he/she is paid accordingly.
- A time statement list can be printed for each employee at the end of the month.

# Time Management

Time recording

Time evaluation

Transfer employee time data to the SAP system

Valuate and account time data



Time  
recording



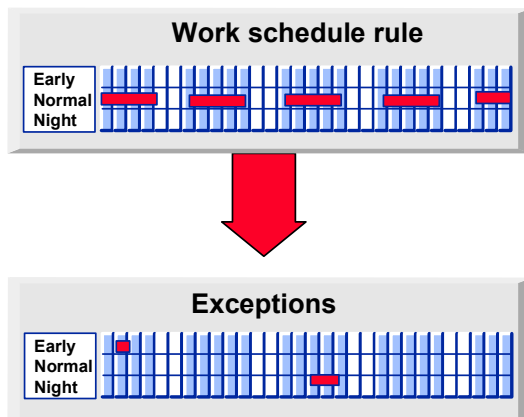
HR report

© SAP AG 1999

- HR Time Management deals with the recording and evaluation of employee time data.
- Times are recorded with a time recording terminal. The recorded times are transferred to Time Management without further valuation.
- The time data is evaluated and remunerated using a report.

## Negative recording

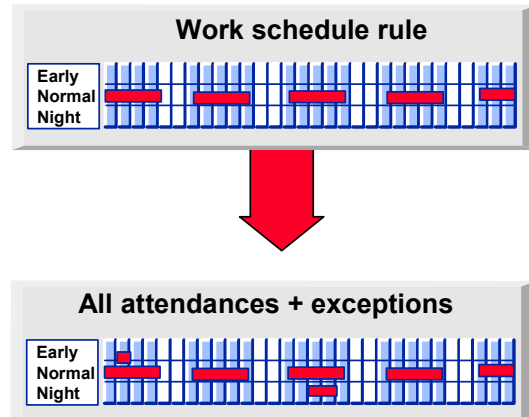
**Record exceptions to the work schedule**



## Positive recording

**Record actual time**

- Automatically
- Manually



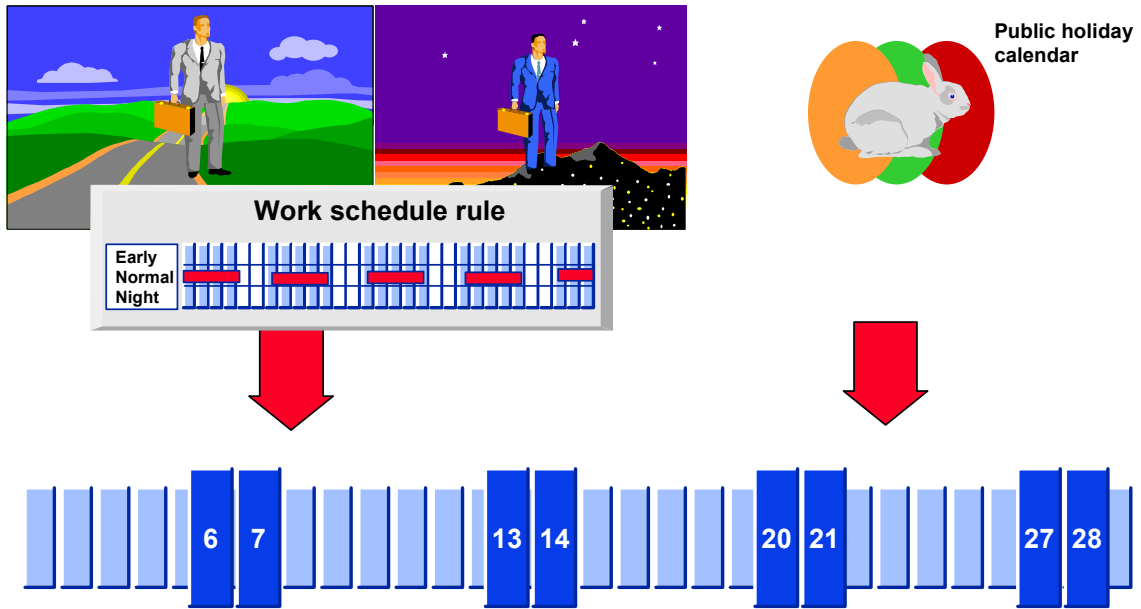
© SAP AG 1999

### ■ There are two methods for recording times:

- Recording exceptions to the work schedule: only time data that is different from the specifications in the employee's work schedule is recorded; for example, sickness, substitution, and vacation.
- Additional recording of actual times: all the employee's attendance times (actual times) are recorded, as well as any exceptions to the work schedule.

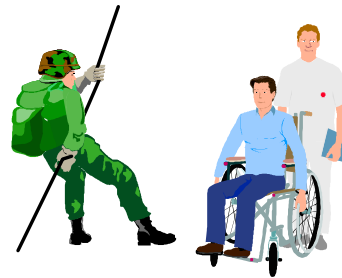
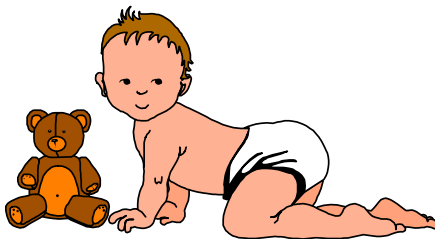
### ■ Actual times can be recorded in two ways:

- Automatically:  
The actual times are recorded in the time recording terminal and are then transferred to the SAP system. They are then processed by a time evaluation program.
- Manually:  
Actual times can be manually entered in the Attendance infotype (2002).



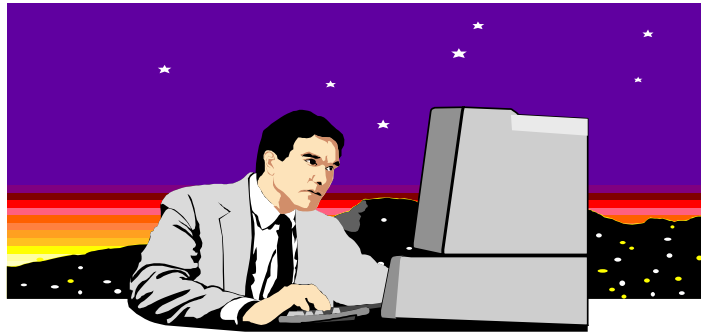
© SAP AG 1999

- To work with the Time Management system, two items are required:
  - A calendar of public holidays
  - A work schedule, which specifies daily and monthly working patterns



© SAP AG 1999

- You can record employee absences in infotype 2001. Employees are considered absent if they do not complete the planned working time specified in their work schedule.
- Absences are grouped into various absence types, which form the subtypes of the Absences infotype (2001):
  - Leave
  - Sickness
  - Family medical leave
  - Military/non-military service, and so on.
- There are a variety of different entry screens in infotype 2001, depending on whether you are recording a general absence, an absence with quota deduction, or a period of sickness or leave.

**Accounting/Logistics data**

- Activity allocation
- Cost assignment
- External services

**Alternative payment**

Premium  
Pay scale group/level  
Extra pay ....

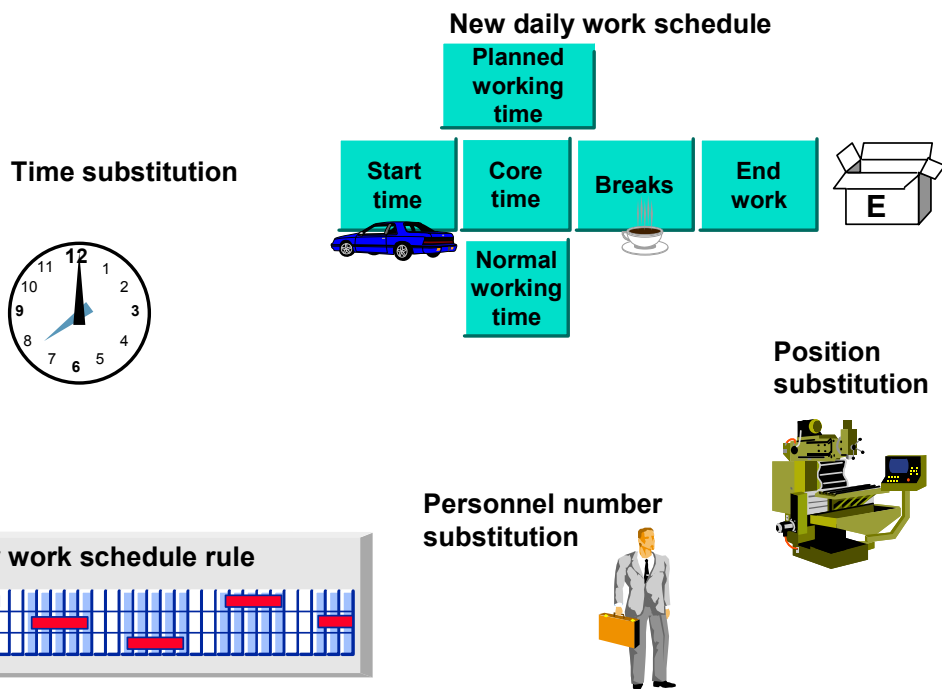
**Cost assignment**

Cost center  
Order  
....

© SAP AG 1999

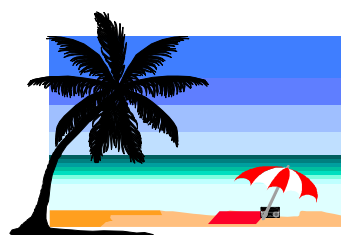
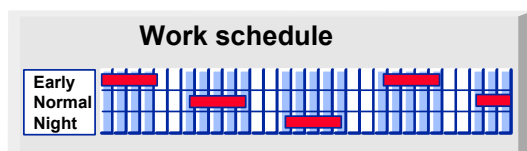
- Overtime is time worked in addition to the planned working time defined in the employee's work schedule rule:
  - Recorded in the Overtime infotype (2005)
  - Calculated automatically if you work with time evaluation
  - Always outside of the employee's planned working time
- Records can be for a period of one or more days.



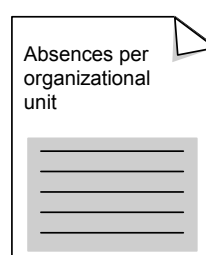
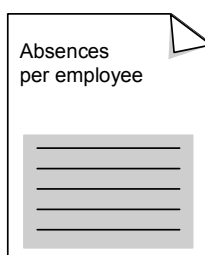
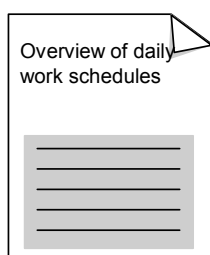
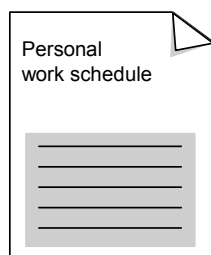
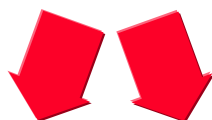


© SAP AG 1999

- Changes to regular working hours are recorded in the Substitutions infotype (2003).
- Standard substitution types include:
  - Substitution with the individual daily work schedule
  - Substitution with the daily work schedule
  - Substitution with the work schedule rule
  - Substitution with the work schedule rule assigned to another employee (personnel number substitution)
  - Substitution with the work schedule rule assigned to another position (position substitution)
- Substitutions represent different working times for an employee. Position substitutions involve a different rate of payment, but the same working time.

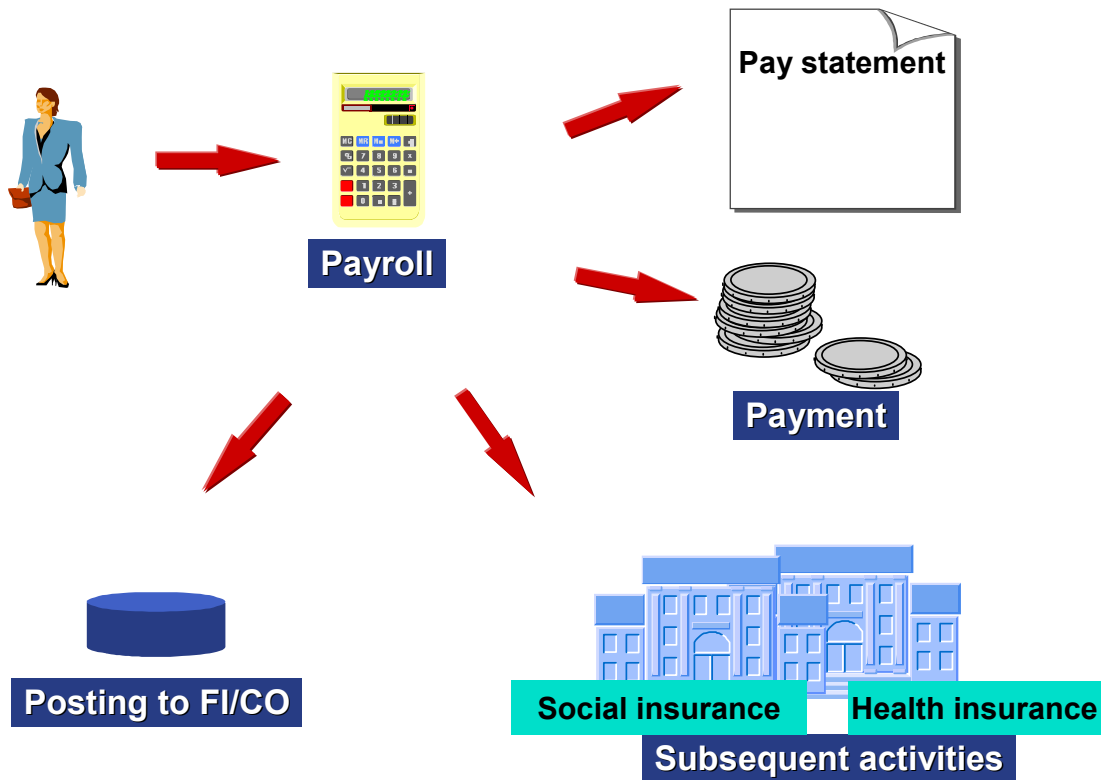


Exceptions to  
work schedule



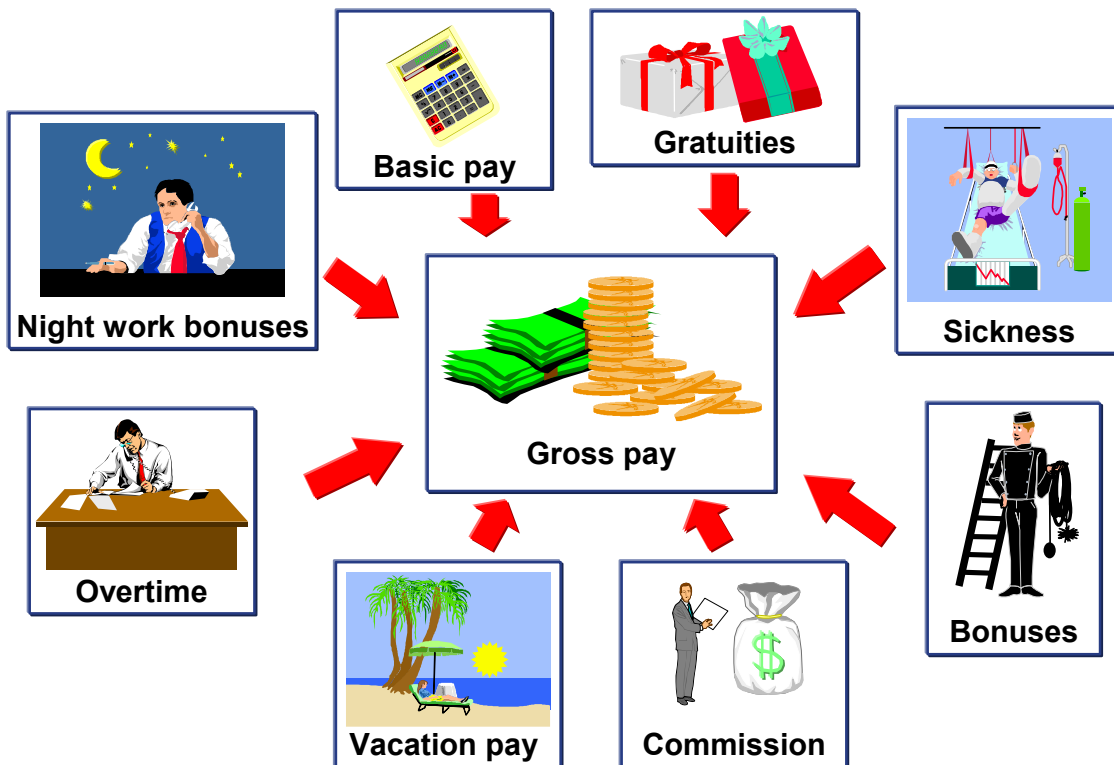
© SAP AG 1999

- You can run various reports in Time Management based on monthly work schedule data and attendance/absence data.



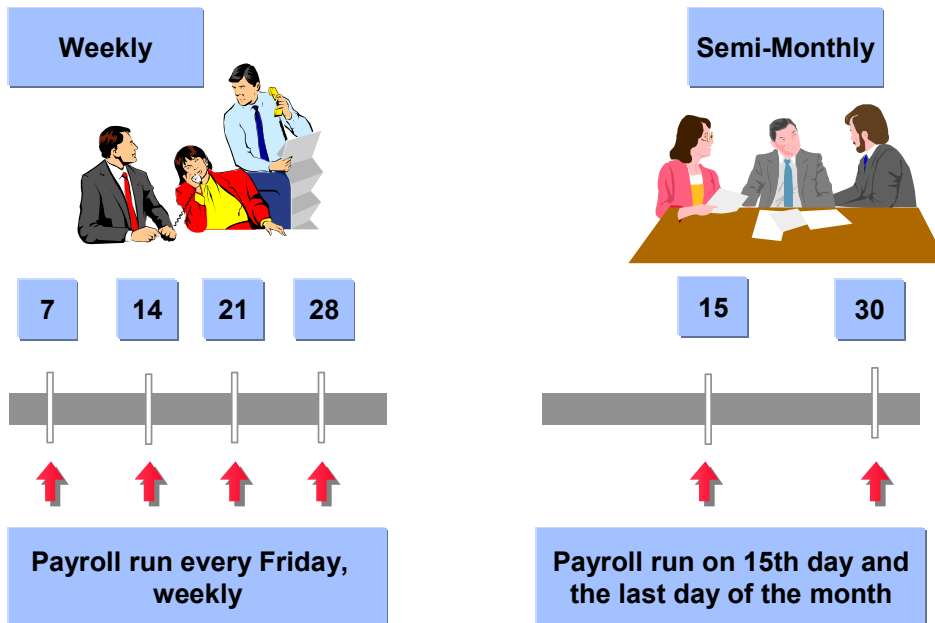
© SAP AG 1999

- An employee's pay data is evaluated in a payroll run. The result is that an employee receives a remuneration statement and the amount is transferred or paid by check.
- After payroll has been run and verified, subsequent activities can be carried out. This includes direct deposit transmissions, tax filings, and posting results to accounting.



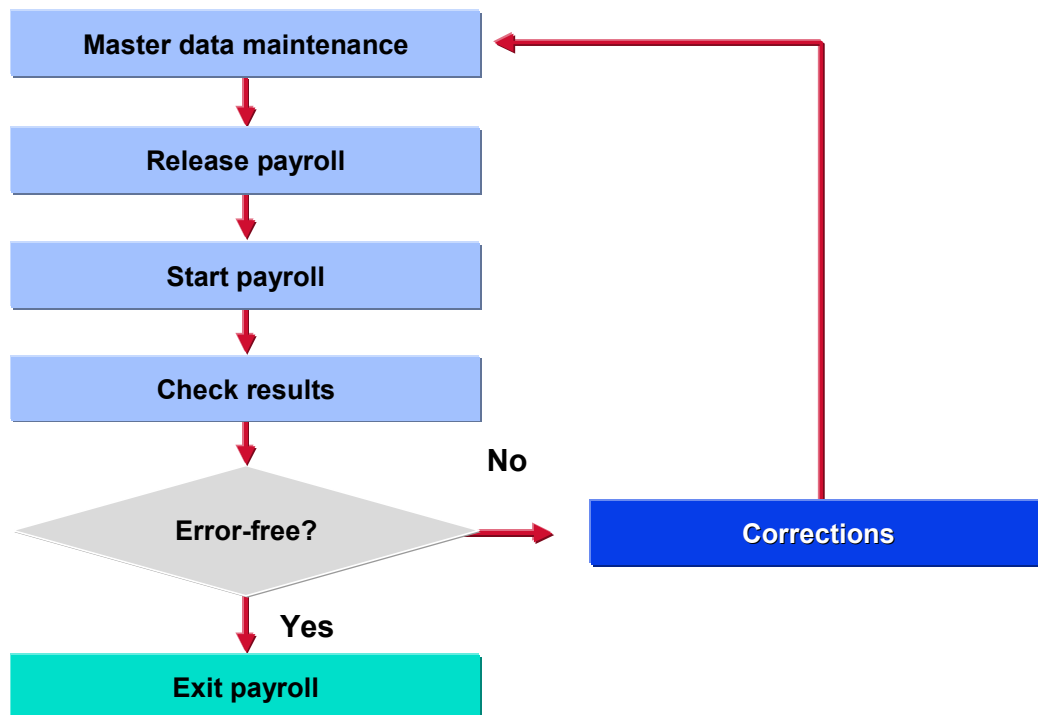
- The amount of an employee's gross pay is calculated on the basis of individual wage types used in a payroll period.
- Payments that may be included in the calculation of gross pay are basic pay, bonuses, vacation pay, Christmas bonuses, gratuities, and so on.
- Deductions might include United Way, a medical or bond deduction, for example. A wage type is a four-digit code that represents an earning or a deduction. Wage types can be entered on-line or generated during the payroll run using rule tables.

**A payroll area is an organizational unit for payroll**



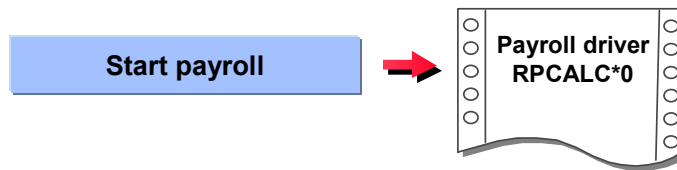
© SAP AG 1999

- The payroll program is usually run separately for different employee groups. A typical example is that of salaried employees and hourly employees, for whom payroll is performed using two different payroll periods. The payroll area determines when payroll is run.
- The payroll area serves two functions, both of which are prerequisites for running the payroll. The payroll area groups together personnel numbers that must be processed in the same payroll run
- The payroll area determines the payroll period.
- Employees for whom payroll is performed at the same time and for the same payroll period are assigned to the same payroll area. Personnel numbers are assigned to a payroll area in the Organizational Assignment infotype (0001).



© SAP AG 1999

- The payroll control record keeps track of each payroll run to ensure that each is processed sequentially.
- When the master data has been maintained, the next step is to release payroll.
- When payroll has been released for a payroll area, you cannot make changes to past or present data that would affect personnel numbers assigned to the respective payroll area. You can still make changes that apply to a future date.
- When a payroll run is started, you are not permitted to make any changes to master or time data that affect the payroll, past or present. The payroll control record ensures that payroll runs and maintenance activities do not conflict.
- If you set the status to Check result, it is not possible to make any changes that affect the payroll, past or present. It is also not possible to make changes to Customizing tables or to the start data for the payroll run. The payroll run is frozen until you release the run for payroll or for corrections.
- The correction status releases the employee's data. You can then enter changes for the payroll period and for the previous periods. If data in the payroll past has been changed, the system automatically triggers a retroactive run for the employees in question.
- You cannot activate this particular parameter until payroll is complete for all personnel numbers in the selected payroll area.



The screenshot shows the "Payroll Driver" selection screen. The title bar is "Payroll Driver". Below the title bar is a toolbar with icons for selection, help, and other functions. The main area is divided into two sections: "Payroll period" and "Selection".

**Payroll period**

Payroll area: U1

☒ Current period

☐ Other period

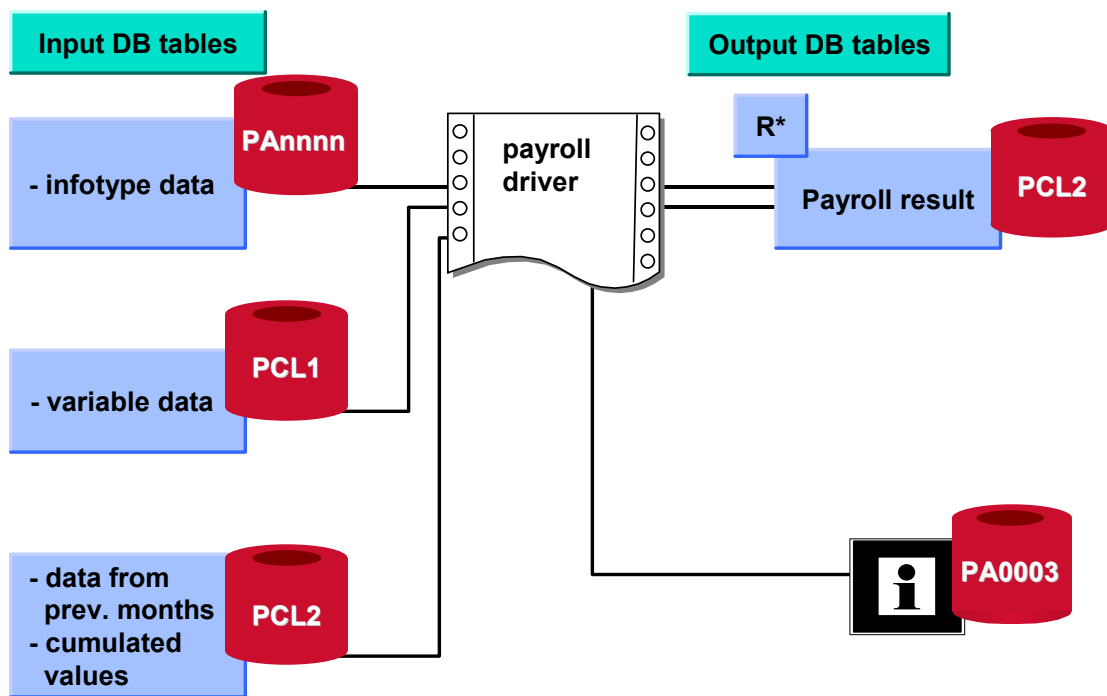
**Selection**

Personnel number: [ ] [ ]

Payroll area: U1 [ ]

© SAP AG 1999

- The payroll area is written directly from the control record to the payroll program. In a productive system, the payroll period is a default value.
- Personnel numbers are selected for the payroll run by entering a payroll area. The current payroll period is the last period accounted plus one.
- The control record contains the last period accounted for each payroll area.



© SAP AG 1999

- Personnel data such as Basic Pay (infotype 0008/master data) and Employee Remuneration Information (infotype 2010/time data) is imported to the payroll program from the infotypes (database table PAnnnn).
- The payroll program maintains the Payroll Status infotype (0003), including retroactive accounting, correction runs, matchcode W, and so on.
- PCL1 Primary data such as time tickets and time results.
- PCL2 Secondary data such as time pairs and payroll results.  
The cluster identifier for the country-specific payroll results is stored in table T500L.
- The name of the cluster for the country-specific payroll results is stored in table T500L.



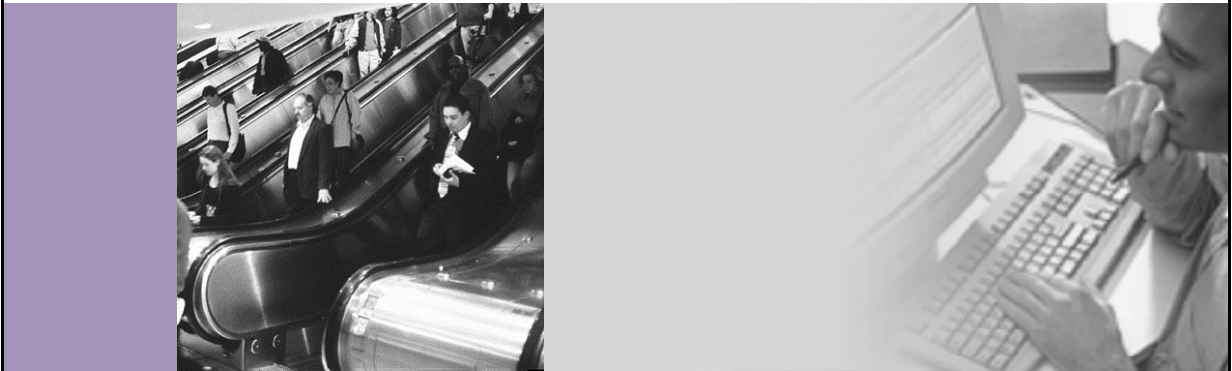
## Cluster display: technical view



© SAP AG 1999

- The RPCLSTxy reports enable you to view different clusters in the database tables PCL1 or PCL2.
- PCL2 contains the following clusters that are relevant to payroll:
  - PS Schema
  - xy Payroll results
  - CU Directory of payroll results
- Cluster RU contains the results for the US payroll run and cluster RD contains the results of the payroll run for Germany.





# HR350

## **Programming in HR**

THE BEST-RUN BUSINESSES RUN SAP



© SAP AG 2003

- SAP R/3 Enterprise
- 2003/Q3
- 50063940

**Copyright 2003 SAP AG. All rights reserved.**

**No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.**

**All rights reserved.**

© SAP AG 2003

## **Trademarks:**

- Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.
- Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.
- IBM®, DB2®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corporation in USA and/or other countries.
- ORACLE® is a registered trademark of ORACLE Corporation.
- UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.
- Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.
- HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- JAVA® is a registered trademark of Sun Microsystems, Inc.
- JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- MarketSet and Enterprise Buyer are jointly owned trademarks of SAP AG and Commerce One.
- SAP, SAP Logo, R/2, R/3, mySAP, mySAP.com, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies.

### Requirements:

- **HR100** (*Essentials of Personnel Administration*)
- **HR120** (*Essentials of Personnel Planning*)
- **BC400** (*ABAP Workbench: Foundation*)
- **BC405** (*Techniques of List Processing*)
- **Experience of programming in ABAP**



### Target group:

- IT staff
- IT administrators

### Duration:

- 5 days



© SAP AG 2003

### Note:

- The training materials do **not** constitute a **teach-yourself program**. They **complement the course instructor's explanations**. Your material includes space for making note of additional information.
- There may not be enough time to complete all exercises during the course. The exercises contain additional examples that are dealt with during the course. Participants can use these examples to consolidate their knowledge once they have completed the course.

### **Contents:**

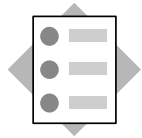
- **Course Goals**
- **Course Objectives**
- **Course Content**
- **Main Business Scenario**



**This course will prepare you to:**

- **Program your own HR reports using logical databases and HR-specific statements**
- **Create customer-specific infotypes**





**At the conclusion of this course, you will be able to:**

- **Use the HR logical databases and the Join and Projection views**
- **Use HR-specific statements**
- **Import and process payroll results**
- **Create customer-specific infotypes**

## Preface

Unit 1	<b>Course Overview</b>	Unit 7	<b>Cluster Database Tables in HR</b>
Unit 2	<b>Data Structures in Personnel Administration</b>	Unit 8	<b>Data Structures in Personnel Planning</b>
Unit 3	<b>The Logical Databases PNP/PNPCE</b>	Unit 9	<b>The Logical Database PCH</b>
Unit 4	<b>Join and Projection</b>	Unit 10	<b>Customer-Specific Infotypes</b>
Unit 5	<b>Specific Commands</b>	Unit 11	<b>HR ABAP Features</b>
Unit 6	<b>Repetitive Structures and List Display</b>		<b>Appendix</b>
			<b>Introduction to HR</b>

© SAP AG 2003

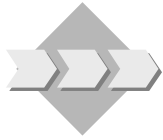
### ■ Note:

In the sample source code contained in these training materials, all ABAP keywords and the names of HR data structures are printed in upper case.

In the solutions for the programming exercises, only the ABAP keywords are written in upper case.

- One part of the appendix contains additional overheads, which the trainer can incorporate into the corresponding units, where appropriate.

Another part contains material for an introduction to HR, which the trainer may use at the start of the course, if necessary.

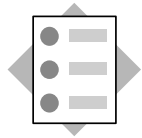


- **All components of the HR system have been implemented at your company.**
- **The Personnel Department in your company requires its own reports and customer-specific infotypes to cover special requirements.**



## Contents

- Infotypes
- Technical data structures of infotypes
- Check and control tables



**At the conclusion of this unit, you will be able to:**

- **Describe the data structures of Personnel Administration**
- **Use the check and control tables of HR infotypes as a source of information**

Course Overview

Cluster Database Tables in HR

 **Data Structures in Personnel Administration**

Data Structures in Personnel Planning

The Logical Databases PNP/PNPCE

The Logical Database PCH

Join and Projection

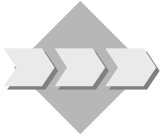
Customer-Specific Infotypes

Specific Commands

HR ABAP Features

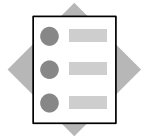
Repetitive Structures and List Display

© SAP AG 2003



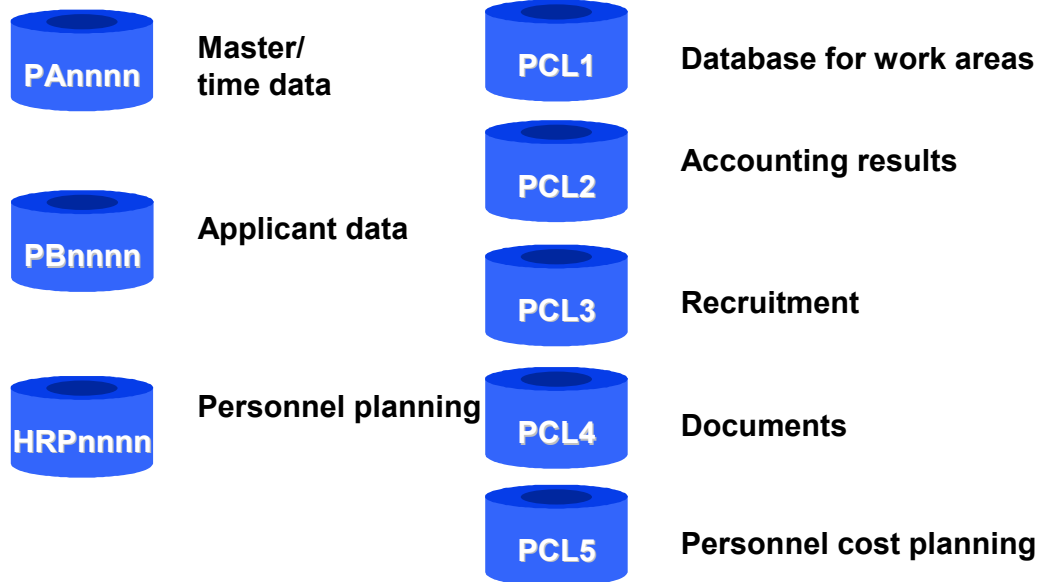
- You want to draw up your own evaluations of data in Personnel Administration. To do this, you must learn about data structures in Personnel Administration.





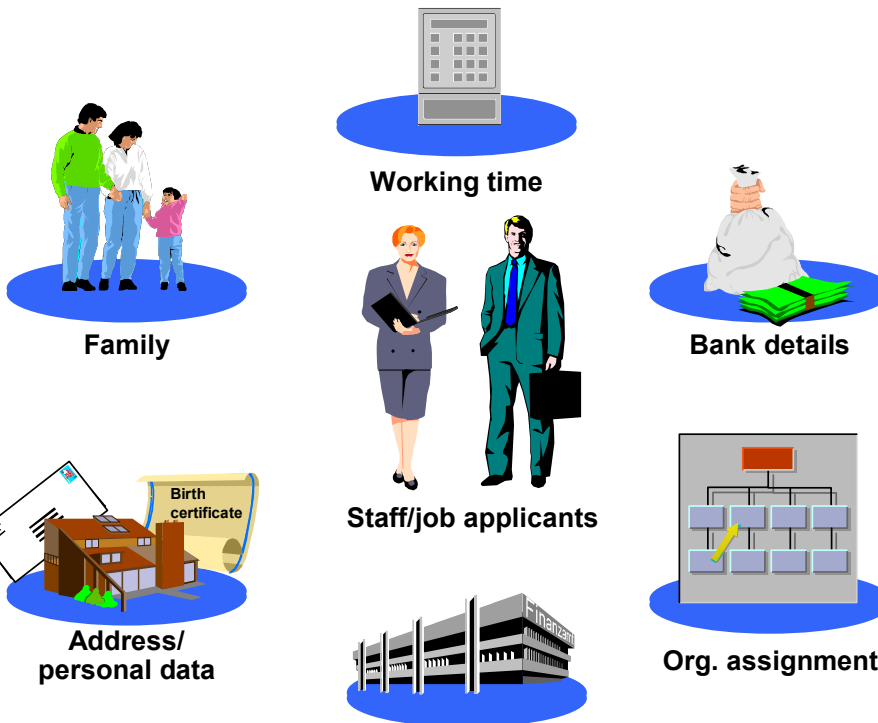
**At the conclusion of this topic, you will be able to:**

- **Describe the concept of infotypes in HR**



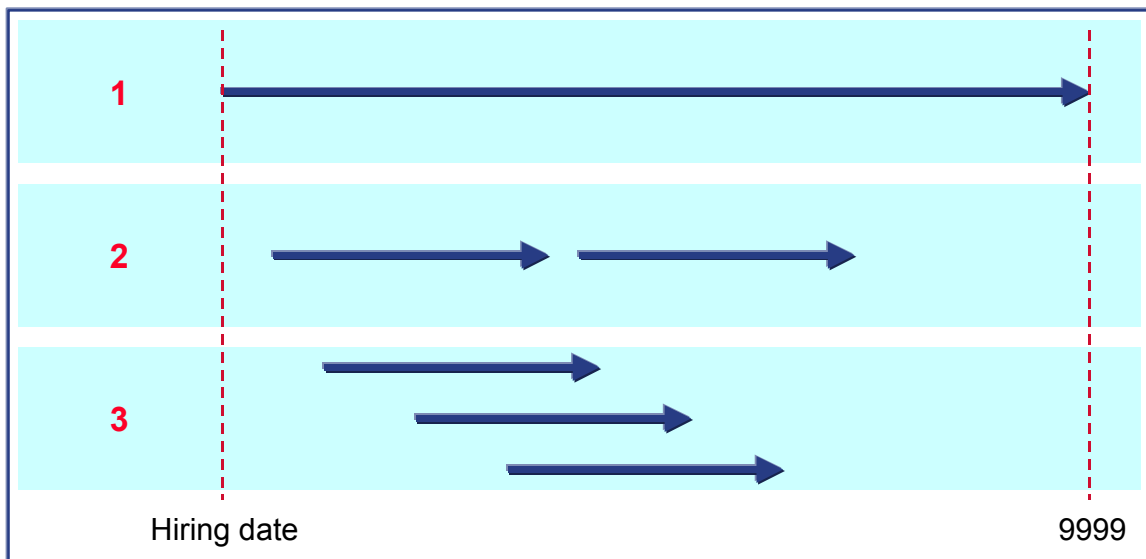
© SAP AG 2003

- Data structures in HR are a combination of master data, transaction data, and results data.
- The master data of Personnel Administration, Time Management, and Personnel Planning are structured in the same way.
- There are also different data clusters, which contain data from specific processing steps, such as time events on cluster PCL1 or accounting results on cluster PCL2.
- The infotypes are grouped into different **number ranges nnnn**:
  - 0000 – 0999 HR master data
  - 1000 – 1999 Personnel Planning
  - 2000 – 2999 Time Management
  - 4000 – 4999 Recruitment
  - 9000 – 9999 Customer-specific enhancements



© SAP AG 2003

- **Infotypes** are information units in the Human Resources Management system. Groups of related data fields are bundled into infotypes. Infotypes structure information, facilitate data entry and allow data to be saved for specific periods. Infotypes serve as templates for users to enter data. From a database point of view, infotypes provide a data structure and a set of coherent data records.
- The infotypes *Actions* 0000, *Organizational Assignment* 0001, *Personal Data* 0002, and *Payroll Status* 0003 are prerequisites for entering a personnel or applicant number.
- Save the infotypes at intervals to build up an infotype history. The system stores a validity period for each infotype record. Therefore, several data records usually exist for each infotype of an employee, and each record has a different validity period.
- You use a **time constraint** to determine how the data records of an infotype react to each other in terms of time.
- Several infotypes have **subtypes**, which break down information further. For example, technically speaking, the address types for infotype *Address* 0006 represent the subtypes.



© SAP AG 2003

- Each infotype and subtype has a time constraint.

- **Time constraint 1:**

Complete saving with no overlaps at any time, that is, one valid data record of the relevant infotype must exist during the duration of the employee's employment at the company.

- **Time constraint 2:**

Incomplete saving without overlaps at any time, that is, of an infotype may exist at a given time. No more than one valid data record of the infotype can exist at any given time.

- **Time constraint 3:**

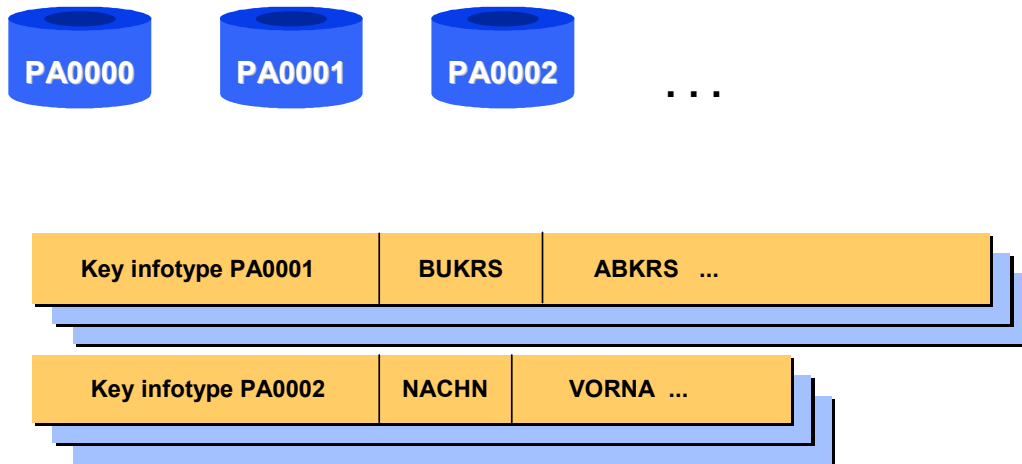
Incompleteness and overlaps permitted, that is, any number of valid data records of an infotype may exist at any given time.



**At the conclusion of this topic, you will be able to:**

- **Explain the technical data structures of the infotypes in Personnel Administration**

The structure of transparent tables:

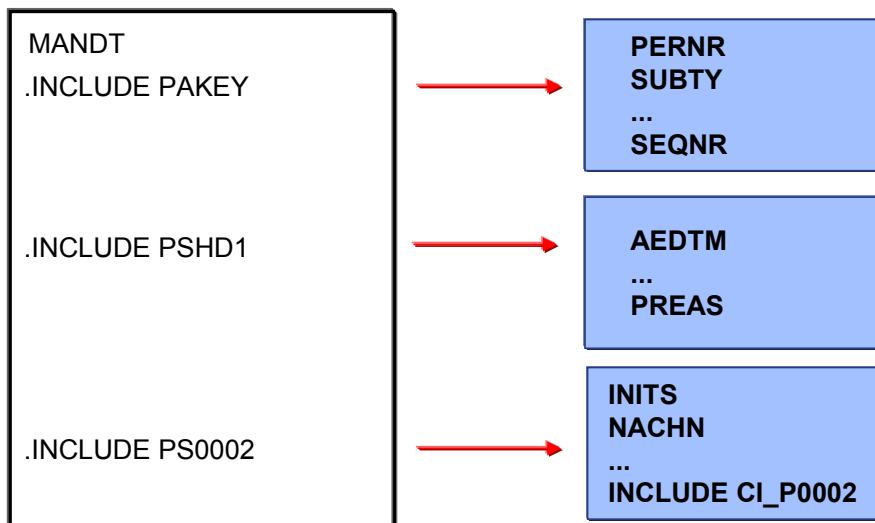


© SAP AG 2003

- The data records of the infotype **nnnn** are saved in the transparent database tables **PAxxxx**.
- The table's **primary key** refers to a minimum quantity of fields in a table, whose value uniquely identifies each data record of the table. In this context, minimum means that after any field is removed from the key, the remaining fields no longer uniquely identify the data records.
- The fields involved in a primary key are referred to as the table's **key fields**. This means that a value combination of the key fields may only occur once in a table.
- A **transparent table** is made up of the primary key and other non-primary keys. As regards infotype tables, non-primary fields include above all the infotype-specific fields, which contain the actual data or information on the infotype.

- Transparent tables *dictionary field definition*

Example: PA0002



© SAP AG 2003

■ Each PAnnnn infotype table and the client (MANDT field) is defined in the ABAP Dictionary using the following three include structures:

- **PAKEY:**  
Contains the infotype table key fields
- **PSHD1:**  
Contains administrative information on the infotype record (amongst other things, the date the last change was made and the name of the user who made the change)
- **PSnnnn:**  
Contains the infotype-specific fields that hold the infotype's actual information and data

**.INCLUDE PAKEY****PERNR****SUBTY****OBJPS****SPRPS****ENDDA****BEGDA****SEQNR**

© SAP AG 2003

- The **PAKEY** structure contains an infotype's primary key. It is the same for all Personnel Administration infotypes and is made up of the following key fields:
- **PERNR**: In a client, the personnel number is the only unique key that identifies an employee. It can be used to display and maintain master data and working time data (infotypes) for an employee.
  - **SUBTY**: Subtypes are subdivisions of infotypes. The subtypes of an infotype may have different time constraints and create their own history.
  - **OBJPS**: The object identification indicator is used to differentiate records with the same infotype, subtype, lock indicator, valid from date, and valid to date. For example: Child number in infotype 0021 *Family Member/Dependents*.
  - **SPRPS**: The lock indicator for HR master data can be used to lock and unlock data records. It can be used to guarantee double control, that is, at least two users are involved in writing an active data record to the database. One of the users creates a locked infotype record. Another user unlocks this record by activating it.
  - **ENDDA**: Valid to date
  - **BEGDA**: Valid from date
  - **SEQNR**: The sequential number differentiates infotype records with the same key and the time constraint '3'. Unlike the personnel object identification, it is automatically assigned by the system.

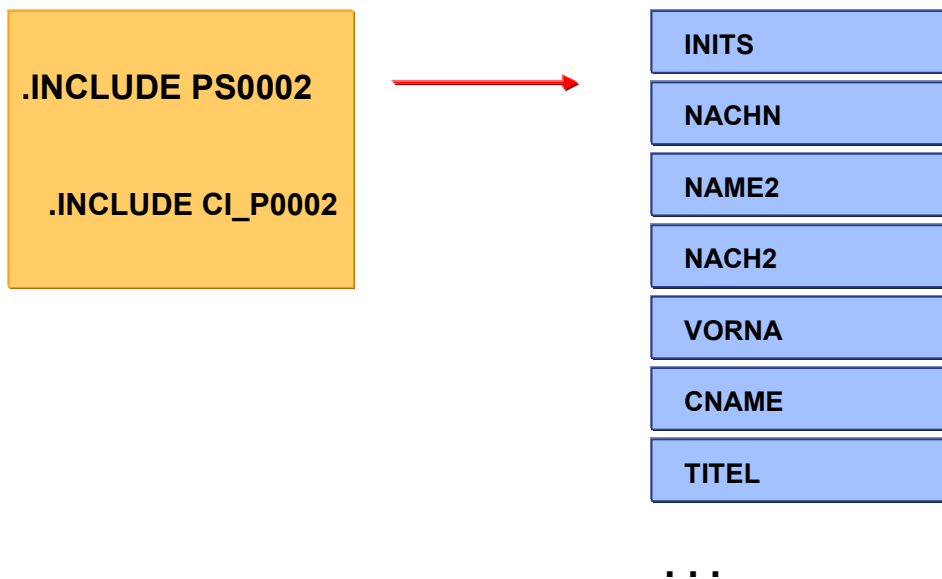


**.INCLUDE PSHD1****AEDTM****UNAME****HISTO****ITXEX****REFEX****ORDEX****ITBLD****PREAS****FLAG1****FLAG2**

© SAP AG 2003

- The structure **PSHD1** is another basic structure that is the same for all Personnel Administration infotypes. Other basic administrative information for each infotype is saved here.
- This information includes the date the record was changed (**AEDTM**), the name of the person making the change (**UNAME**), and information on whether the administrator created plain text for this infotype (**ITXEX**).

Example: PA0002



© SAP AG 2003

- The structure **PSnnnn** contains the infotype-specific data fields, that is, the data fields that contain the actual infotype-specific characteristics.
- Each Personnel Administration infotype contains an include **CI\_Pnnnn** in the data structure PSnnnn. Customer-specific fields can be added to this include as enhancements to an infotype. In the standard SAP system, the include CI\_Pnnnn is empty.

## DB structure PAnnnn using infotype 0002 as an example



MANDT	PAKEY	PSHD1	PS0002
-------	-------	-------	--------

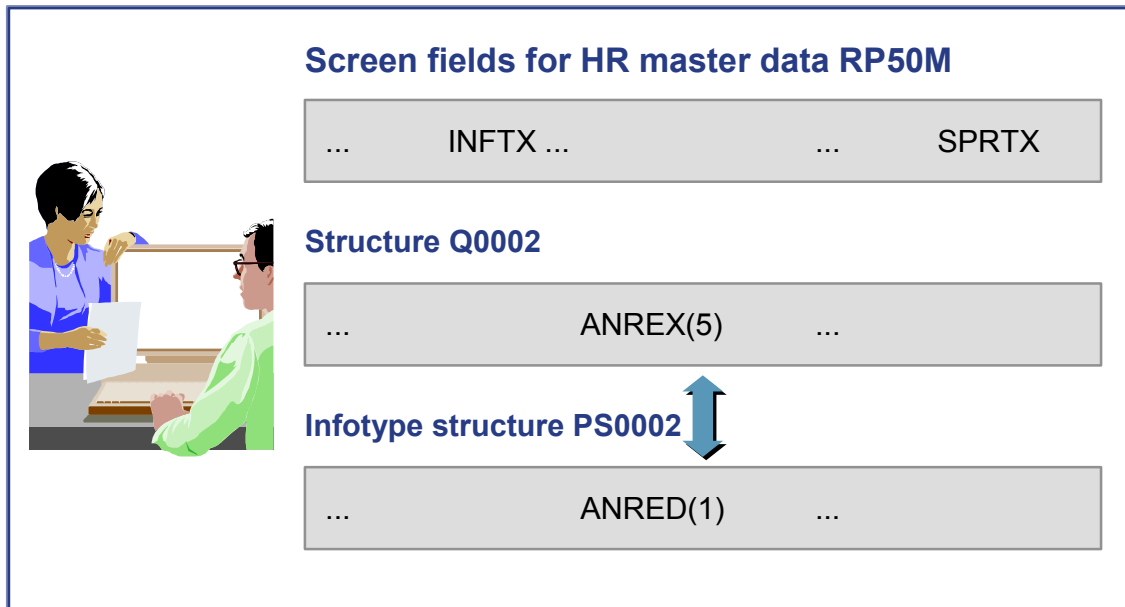
## Infotype structure Pnnnn using infotype 0002 as an example



PSKEY	PSHD1	PS0002
-------	-------	--------

© SAP AG 2003

- During interface definition, the logical structure **Pnnnn** is used between programs and between programs and screens.
- The key **PSKEY** contains the key fields of the structure **PAKEY**, and an additional field **INFTY** for the infotype number **nnnn**. The client field is not required in this structure.
- The structure **PSnnnn** is also included by the structure **Pnnnn** so that the transparent table **PAnnnn** and the logical structure **Pnnnn** are always consistent with each other.



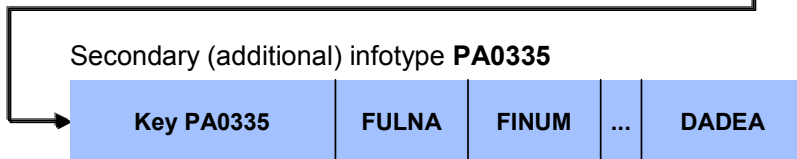
© SAP AG 2003

- Fields (such as header lines) that are displayed for all infotypes are stored in the structure RP50M.
- The structure **Qnnnn** also exists for several infotypes. It contains screen fields for data entry. An employee's form of address, for example, is stored with a numeric key in the field PA0002-ANRED. The screen, however, must enable the form of address itself to be displayed and entered, not its numeric key. For this reason, the additional field Q0002-ANREX is used.

- **Infotype view consisting of a primary and secondary (additional) infotype**

Primary infotype **PA0021**

Key PA0021	FAMSA	FGBDT	...	EMRGN
------------	-------	-------	-----	-------



**Tables T582V/T582W:** View definition - assignment of additional infotype to primary infotype

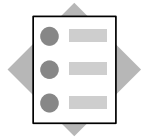
© SAP AG 2003

- Many country versions require country-specific fields that are attached to all international infotypes, such as *Organizational Assignment*, *Family Member/Dependents*.
- To prevent international infotype structures being crammed with country-specific fields, these fields have been moved to **additional infotypes**.
- Whereas international infotypes ("primary infotype") have the value *I*, additional infotypes have the value *Z* in the *Infotype/View* field in the table T777D.
- The primary infotype and the additional infotype are maintained together on one screen in an **infotype view** (defined using the tables T582V and T582W). Then the infotypes are saved, the data you are maintaining is distributed across two infotype records (one primary and one additional infotype record) with an identical infotype key.
- The infotype menu only contains the international primary infotype. The additional infotype can only be maintained with the primary infotype, and not on its own. All infotype characteristics such as the time constraint are only defined on the primary infotype.
- When a personnel number is being created, the attribute IVWID is used to assign a fixed view indicator in the infotype *Payroll Status* 0003. The secondary infotypes are selected using this view indicator. This view indicator must NEVER be changed, for obvious reasons.

Element	Definition	Included	Changeability
<b>PAKEY</b>	<b>Structure</b>	-	<b>No</b>
<b>PSHD1</b>	<b>Structure</b>	-	<b>No</b>
<b>PSnnnn</b>	<b>Structure</b>	<b>CI_Pnnnn</b>	<b>PS0nnn-PS8nnn: Only CI_Pnnnn PS9nnn: Yes</b>
<b>Pnnnn</b>	<b>Structure</b>	<b>PSKEY PSHD1 PSnnnn</b>	<b>P0nnn-P8nnn: No P9nnn: Only PS9nnn</b>
<b>PAnnnn</b>	<b>Transparent table</b>	<b>MANDT PAKEY PSHD1 PSnnnn</b>	<b>PA0nnn-PA8nnn: No PA9nnn: Only PS9nnn</b>
<b>CI_Pnnnn</b>	<b>Structure</b>	-	<b>Yes</b>

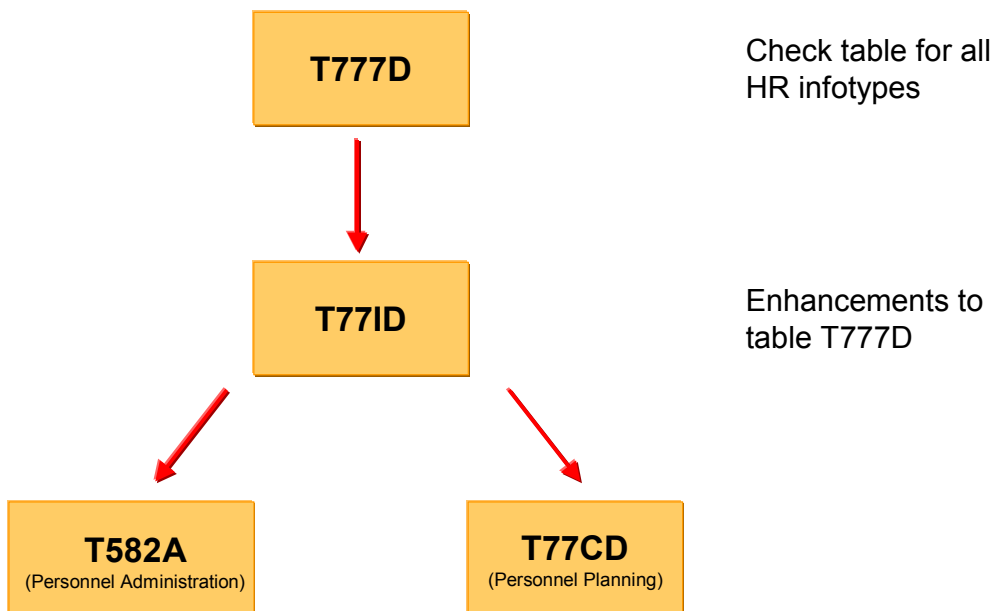
© SAP AG 2003

- The table displayed above provides an overview of the data structures and transparent tables that belong to an infotype in Personnel Administration.



**At the conclusion of this topic, you will be able to:**

- **Use the technical information contained in the check and control tables of HR infotypes**



© SAP AG 2003

- The table *Infotypes - Dialog/Database Assignment* (T777D) is the key check table (domain INFOTYP) for all HR infotypes (administration and planning infotypes). This table contains the names of all infotype-dependent repository objects (such as tables, structures, and programs).
- The table *Infotypes - Enhancements to T777D* (T77ID) supplements the table T777D that exists for reasons of memory space.
- The table *Infotypes - Customer-Specific Settings* (T582A) is used to make customer-specific settings for administration infotypes.
- The table *Infotypes - Customer-Specific Settings* (T77CD) can be used to make customer-specific settings for planning infotypes.
- You can use the report RHT777DCHECK to check the consistency of entries in the specified tables.



- Infotype control

Table T777D

Infotype	Structure	DB Table	DB Table Table Infotype
0000	P0000	PA0000	
0001	P0001	PA0001	
...			
1000	P1000	HRP1000	
1001	P1001	HRP1001	
1002	P1002	HRP1002	HRT1002
...			
9998	P9998	PA9998	
9999	P9999	HRP9999	

© SAP AG 2003

- The table *Infotypes - Dialog/Database Assignment* (T777D) is automatically maintained using the tool for creating new infotypes. Do not edit this table manually.
- In addition to the transparent tables **PA#####** and **HRP#####**, each infotype in the Human Resources Management system has a **logical structure P#####** (stored in T777D-PPNNN).
- Logical structures are used to define interfaces between programs and between screens and programs. No data records exist on the database for a logical structure.
- To avoid direct database accesses, these logical structures P##### are primarily used in Human Resources Management programs.

**Table T777D**

Infotype	Structure	Infotype-Spec. Fields	Prim. Table Ext. Infotype
0000		PS0000	
0001		PS0001	
...			
1000		HRI1000	
1003		HRI1003	
...			
1611		HRI1611	V_T5U27
1612		HRI1612	V_T5U26
...			
9998		PS9998	
9999		HRI9999	

© SAP AG 2003

- The table *Infotypes - Enhancements to T777D* (T777ID) only supplements the *Infotypes - Dialog/Database Assignment* table T777D, for reasons of memory space.
- This table includes the structure PSnnnn or HRInnnn for infotype-specific fields for each infotype.

- Control applicant infotypes

**Table T777D**

Infotype	DB Table	DB Table Applicant Infotype
0000	PA0000	PB0001
0001	PA0001	PB0002
0002	PA0002	
...		PB0006
0006	PA0006	PB0007
0007	PA0007	
...		PB4000
4000		PB4001
4001		PB4002
4002		

© SAP AG 2003

- Infotype records for applicant infotypes are stored in the transparent tables **PBnnnn**, which are structured in the same way as PAnnnn tables.
- The following types of applicant infotypes exist:
  - Infotypes that can be maintained for employees and applicants (for example, 0001, 0002, 0006, and 0007). Here, there is a database table **PAnnnn** for employee data and a database table **PBnnnn** for applicant data.
  - Infotypes that may only be maintained for applicants (applicant infotypes only, with infotype numbers from 4000 to 4999). Here, there is only one database table solely for applicant data (**PB4nnn**).

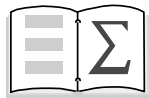
- Subtype control

Table T777D

Infotype	Subtype Field	Subtype Table	Subtype Text Tab.	TC Table
0006	ANSSA	T591A	T591S	T591A
...				
0008	SUBTY	T591A	T591S	T591A
0009	BNKSA	T591A	T591S	T591A
0010	LFDNR	T591A	T591S	T591A
0011	LGART	T512Z	T512T	T591B
...				
0019	TMART	T531	T531S	
...				
0022	SLART	T517T	T517T	

© SAP AG 2003

- *Subtypes* are used to further break down *infotypes*, for example, to further classify infotype 0006 *Addresses* into main address, second address, business address, and so on.
- In the table T777D, the following fields are used to define subtypes:
  - **Subtype field:** Here, the field is in the infotype structure that contains the subtype. Usually this is the field *SUBTY*. However, it may also be a field other than *SUBTY* and contain the value of the subtype (for example, *ANSSA* = address type in the infotype 0006).
  - **Subtype table:** This contains the check table that contains all allowed subtype values for each infotype. This is usually the table *T591A*. However, the subtype values may also be stored in another table - depending on the subtype field - for example, the table *T512Z* for infotype 0011 *Ext. Bank Transfers* (permissible wage types for each infotype), as the wage type (field *LGART*) is also the subtype.
  - **Subtype text table:** This lists the text table for the subtype table, that is, the table with the names for the individual subtype values. This is usually the table *T591S*, if the table *T591A* is used as a subtype table. If another subtype table is used (for example, *T512Z*), the corresponding text table (for example, *T512T*) is stored here.
  - **Time constraint table:** If the time constraint is defined based on the subtype, the corresponding time constraint table appears here (for example, *T591A/T591B*).



**You are now able to:**

- **Describe the data structures of Personnel Administration**
- **Use the check and control tables of HR infotypes as a source of information**



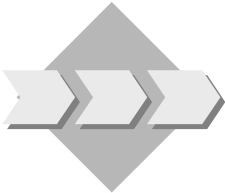


## Unit: Data Structures in Personnel Administration



At the end of this exercise, you will be able to:

- Describe the data structures of Personnel Administration
- Use the check and control tables of HR infotypes as a source of information



You want to draw up your own evaluations of Personnel Administration data. To do this, you must learn more about data structures in Personnel Administration.

- 1-1 The fields SPRPS and OBJPS are located in the key of Personnel Administration infotypes. Write down the functions of both fields.

1-1-1 **SPRPS:**

---

---

1-1-2 **OBJPS:**

---

---

- 1-2 What tasks do the structures Qnnnn and CI\_Pnnnn fulfil?

1-2-1 **Qnnnn:**

---

---

1-2-2 **CI\_Pnnnn:**

---

---

- 1-3 Several infotypes have subtypes. If this is the case, the infotype's time constraint is on the subtype.

1-3-1 How do you determine which time constraint is assigned to the subtype of an infotype?







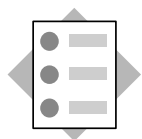
## Unit: Data Structures in Personnel Administration

- 1-1 The fields SPRPS and OBJPS are located in the key of Personnel Administration infotypes. Write down the functions of both fields.
- 1-1-1 **SPRPS:**  
If this field is not blank, the infotype record is locked. It needs to be unlocked to be activated.
- 1-1-2 **OBJPS:**  
This field is used to differentiate infotype records with the same infotype, subtype, block indicator, valid from date, and valid to date.
- 1-2 What tasks do the structures Qnnnn and CI\_Pnnnn fulfil?
- 1-2-1 **Qnnnn:**  
The structure Qnnnn contains screen fields with different lengths than the corresponding database fields.
- 1-2-2 **CI\_Pnnnn:**  
The customer-specific fields of a standard infotype are grouped in the structure CI\_Pnnnn.
- 1-3 Several infotypes have subtypes. If this is the case, the infotype's time constraint is on the subtype.
- 1-3-1 To establish the time constraint of the infotype's subtype, determine which time constraint table is assigned to the infotype in the table T777D. View the table.



## **Contents:**

- **Functions of the logical database**
- **Data retrieval**
- **Processing of infotypes**
- **Period-based data**
- **Person selection and data selection**
- **Sort order**
- **Setting up the selection screen**
- **Selection view for free delimitations**
- **The logical database PNPCE**



**At the conclusion of this unit, you will be able to:**

- **Explain the functions and data structures of the logical database PNP**
- **Process period-based data in a report**
- **Use the screening criteria to select data and persons**
- **Set up the standard selection screen**
- **Create a selection view for free delimitations**
- **Use the logical database PNPCE in PNP mode**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration

Data Structures in Personnel Planning

 The Logical Databases PNP/PNPCE

The Logical Database PCH

Join and Projection

Customer-Specific Infotypes

Specific Commands

HR ABAP Features

Repetitive Structures and List Display

© SAP AG 2003

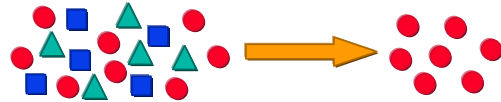


- **The Personnel Department in your company requires its own reports to evaluate HR master data.**
- **As a member of the IT team, it is your responsibility to program these reports.**
- **You would like to make optimal use of the functions offered by the logical databases in HR.**

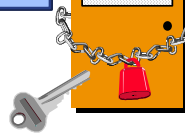
## 1. Data retrieval



## 2. Screening



## 3. Authorization check



© SAP AG 2003

- To minimize the amount of programming required, it often makes sense to use logical databases when generating reports. Reports are special programs that provide selection screens and access databases. The logical database PNP consists of the database driver **SAPDBPNP** and is activated using the report attributes. It fulfils three main functions:
  - **Data retrieval:**  
HR data is written for each employee to the main memory, where it can be processed.
  - **Screening:**  
A selection screen enables you to select employees in accordance with organizational criteria. For example, you can select all of the hourly wage earners of a particular personnel subarea.
  - **Authorization check:**  
The implicit authorization check is another advantage of the logical database. Personnel data is frequently confidential and it does not make sense to allow all programs carry out the authorization check.

**1. Create data structures for infotypes**

```
TABLES: PERNR.  
INFOTYPES: 0001,  
            0002,  
            0007.
```



"Organizational Assignment"  
"Personal Data"  
"Planned Working Time"

**2. Fill the data structures with the infotype records**

```
GET PERNR.
```

© SAP AG 2003

- The structure **PERNR** must be declared with the **TABLES** statement.
- The **INFOTYPES nnnn** statement is used to create an internal table with the name Pnnnn for each declared infotype.
- When the **GET** event occurs, the tables of declared infotypes are filled with all of the records that exist for a personnel number. The field **PERNR** in the structure **PERNR** contains the personnel number. For purposes of processing, you can access the **PERNR-PERNR** field. This field contains the formatted name of the employee.
- **Note:**  
When the **GET** event occurs, all internal infotype tables are filled. Therefore, only declare the infotypes you wish to process. You can, however, use the enhancement **MODE N** for the statement **INFOTYPES** to suppress the filling of internal tables.



```

TABLES: PERNR.
INFOTYPES: 0000,      "Actions
            0002,      "Personal Data
            0006,      "Addresses
            ....

GET PERNR.
  LOOP AT P0002 WHERE ENDDA GE PN-BEGDA
              AND  BEGDA LE PN-ENDDA.
    WRITE...
  ENDLOOP.
    
```

© SAP AG 2003

- When the GET PERNR event occurs, the infotype records are imported to internal tables Pnnnn (for example, P0006 for infotype 0006). These tables may then be processed in a **LOOP-ENDLOOP** loop.
- All infotype records whose validity period overlaps the period selected in the selection screen (PN-BEGDA and PN-ENDDA) by at least one day are placed one after the other in the header of the respective infotype table Pnnnn.
- For example, if you choose the current year in the selection screen, PN-BEGDA contains the first day of the year and PN-ENDDA contains the last day of the year. If you do not enter any data for the period in the selection screen, PN-BEGDA contains the low date (January 01, 1800) and PN-ENDDA contains the high date (December 31, 9999).
- **Note:**  
Infotypes with the **time constraint 3** must be processed in this way.

```
TABLES: PERNR.  
INFOTYPES: 0001,      "Actions  
            0002,      "Personal Data  
            0006,      "Addresses  
            ....  
  
GET PERNR.  
  PROVIDE * FROM P0002  
    BETWEEN PN-BEGDA AND PN-ENDDA.  
  WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- Infotypes with the time constraint 1 or 2 may be processed with a **PROVIDE-ENDPROVIDE** loop.
- All infotype records whose validity period overlaps the period selected in the selection screen (PN-BEGDA and PN-ENDDA) by at least one day (BETWEEN-AND) are placed one after the other in the header of the respective infotype table Pnnnn.
- **Note:**  
If you have entered period dates in the selection screen, the contents of the fields Pnnnn-BEGDA and Pnnnn-ENDDA are also reset in the **header** of the infotype table Pnnnn. For example, if the date in Pnnnn-ENDDA is after the date in PN-ENDDA, Pnnnn-ENDDA is given the value from PN-ENDDA.

```
TABLES: PERNR.  
INFOTYPES: 0000,      "Actions  
            0002,      "Personal Data  
            0006,      "Addresses  
            ....  
  
GET PERNR.  
  PROVIDE * FROM P0006  
    BETWEEN PN-BEGDA AND PN-ENDDA  
    WHERE P0006-SUBTY = '1'.  
  WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- If the infotype to be maintained has subtypes, it must be processed with the subtype. This is brought about by the **WHERE** condition in the PROVIDE statement. In the above example, processing of the infotype *Addresses* (0006) is restricted to subtype 1 *Permanent Address*. This prevents the intervals from overlapping. This is required as the PROVIDE processing only occurs correctly for infotype records with periods that do not overlap.

```
TABLES: PERNR.  
INFOTYPES: 0002,      "Personal Data  
            0015,      "Additional Payments  
            ....  
  
GET PERNR.  
CHECK LINES( P0015 ) GT 0.  
PROVIDE * FROM P0002  
    BETWEEN PN-BEGDA AND PN-ENDDA.  
    WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- Frequently, you only want to display employees with data of a specific infotype.
- You can use the **CHECK LINES( *Pnnnn* )** statement to determine the number of filled lines of an infotype table. The subsequent statements are only executed if the condition is fulfilled.

```

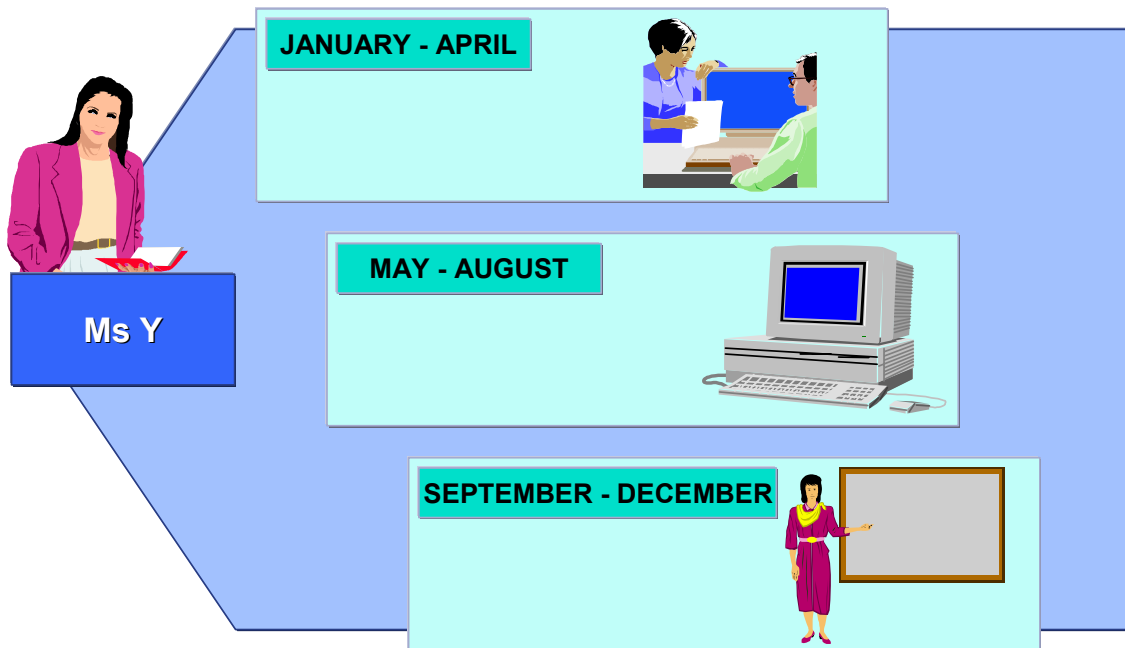
GET PERNR.
  LOOP AT P0002 WHERE ENDDA GE PN-BEGDA
            AND  BEGDA LE PN-ENDDA.
    WRITE...
  ENDLOOP.
or:
  PROVIDE * FROM P0002
    BETWEEN PN-BEGDA AND PN-ENDDA.
  WRITE...
  ENDPROVIDE.

  PROVIDE * FROM P0006
    BETWEEN PN-BEGDA AND PN-ENDDA
    WHERE P0006-SUBTY = '1'.
  WRITE...
  ENDPROVIDE.

END-OF-SELECTION.
    
```

© SAP AG 2003

- HR data is processed in two nested loops:
- A GET PERNR loop for all of the personnel numbers selected. It is concluded implicitly by the next event, for example, END-OF-SELECTION.
- Subordinate loops per infotype, for the processing of all infotype records for the selected personnel number.
- With this form of processing, you should note that the data for the infotypes *Personal Data* (0002) and *Addresses* (0006) is listed sequentially and is not linked.



© SAP AG 2003

- Infotype data is period-based, in other words, it is only valid for specific periods. For this reason, each record has a valid from date and a valid to date.
- This example shows the jobs that an employee has performed over the course of a year.
- The decision on how to retrieve data is made for each individual infotype.

**Data  
selection**



**Org.  
assignment**

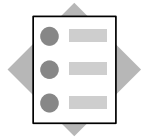


**Org.  
assignment**



© SAP AG 2003

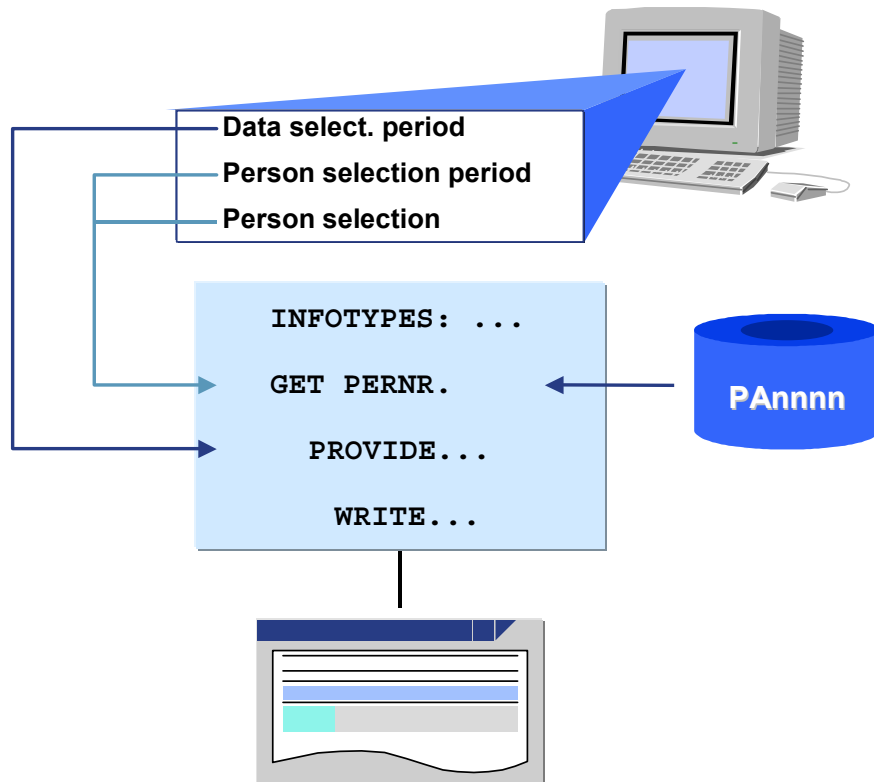
- Both types of data selection are based on the data selection period entered in the selection screen.
- Processing with LOOP-ENDLOOP and PROVIDE-ENDPROVIDE provides data for a specific period, that is, according to the data selection period.
- Frequently, only the most recent or oldest infotype record is required from the selected period, rather than all of the infotype records. The RP\_PROVIDE\_FROM\_LAST and RP\_PROVIDE\_FROM\_FRST macros can be used in such situations.



**At the conclusion of this topic, you will be able to:**

- **Use the screening criteria**
- **Define default values for the selection screen in your report**
- **Sort personnel numbers according to organizational criteria**
- **Set up the standard selection screen to suit a user's needs**
- **Create a selection view for free delimitations**





© SAP AG 2003

- When data is screened, a distinction is made between two types of selection: Person selection, and the data and person selection period.
- The system uses the person selection and the person selection period to determine which employees correspond to the selection criteria, for example, the monthly wage earners for a specific personnel subarea.
- The data selection period specifies the period for which the data contained in the infotype tables is evaluated by the report.
- The selection screen enables you to change the sort sequence for HR records.

Period

☒ Today
 ☐ Current month
 ☐ Current year

☐ To current date
 ☐ From today

☐ Other period

Data select. period  to

Person selection period  to

Payroll period

GET PERNR.

PROVIDE \* FROM Pnnnn  
 BETWEEN PN-BEGDA  
 AND PN-ENDDA.  
 ...  
 ENDPROVIDE.

© SAP AG 2003

- The PN structure (defined in DBPNPCOM) that is available in the report is populated from the selection screen. The data from the start and end dates of the **data selection period** is transferred to the fields PN-BEGDA and PN-ENDDA. If you use a data selection period, infotype data records are entered in the PROVIDE loop if their validity periods overlap with the period selected by at least one day.
- The start and end dates of the **person selection period** that are entered by the user are available in the fields PN-BEGPS and PN-ENDPS. This interval is the criterion used for selecting the personnel numbers to be processed. All personnel numbers that fulfil the selection criteria specified (see overleaf) at any given time in the person selection period entered by the user, are used in processing.
- If the period is specified using radio buttons, the fields PN-BEGDA/ENDDA and PN-BEGPS/ENDPS contain the same period. In the example displayed above, *Today* is set as the key date. In this case, the four fields contain the system date.
- Note that all of the data records for an infotype are available in the corresponding internal table for GET PERNR. In other words, data selection does not restrict the ability to read from the database.

## Selection

Personnel number



Employment status



Company code



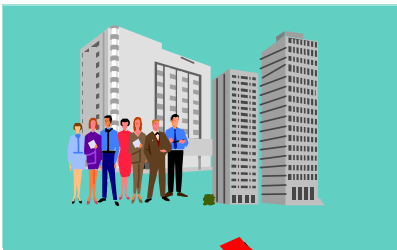
Payroll area



Pers.area/subarea/cost center



Employee group/subgroup



GET PERNR.





© SAP AG 2003

- The objective and result of person selection is a group of employees for whom an evaluation is created.
- You can select persons by entering the following criteria, for example:- A range of personnel numbers- Characteristics of the organizational assignment- The status of the employee, for example, whether s/he is active or inactive
- All the criteria that the user enters here must be fulfilled by the personnel numbers to be selected in the specified person selection period. The data from the logical database is only made available to your report when the event GET PERNR occurs if these criteria are fulfilled.

Period

☐ Today
 ☐ Current month
 ☒ Current year
 ☐ To current date
 ☐ From today
  to 
☐ Other period

Selection

Personnel number    
 Payroll area  

## INITIALIZATION.

```

PNPTIMED          = 'Y' .
PNPABKRS-LOW      = 'D1' .
PNPABKRS-HIGH     = 'D2' .
PNPABKRS-OPTION   = 'BT' .
PNPABKRS-SIGN     = 'E' .
APPEND PNPABKRS.
    
```

© SAP AG 2003

- When the INITIALIZATION event occurs, you can set default values for the selection screen.
- Default radio buttons for selecting the period can be set by transferring one of the following values to the field PNPTIMED:
  - D = Today (key date)
  - M = Current month
  - Y = Current year
  - P = Up to today (from past to current date)
  - F = As of today (from the current date into the future)
- The person selection fields are defined in the include program **DBPNPSEL**. They are internal tables that must be filled using APPEND.

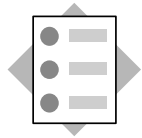


...  
GET PERNR.  
...

PerNo.	Pers.Area	Name
1909	1000	Sam Hawkins
1910	1000	David Lindsay
1899	1100	Karl May

© SAP AG 2003

- The standard sort sequence is in ascending order by personnel number.
- You can use the sort function to sort an evaluation using organizational criteria.
- For example, you can use the sort function to create a hierarchical list according to personnel subarea and administrator, or to list employees in alphabetical order.
- The sort function can be used for all of the fields in infotype 0001 *Organizational Assignment*. You can also determine the sort sequence.
- Different sort types are available for evaluations of specific periods.
- If data is selected by matchcode, personnel numbers are sorted by matchcode sequence.



**At the conclusion of this topic, you will be able to:**

- **Use the logical database PNPCE in PNP mode**

- **Employee as person with multiple employment with several employment contracts**
- **Separate personnel number for each contract**
- **Assignment of individual personnel numbers to a central person (object type CP)**
- **Additional external person ID in the infotype 0709 *Person ID***

© SAP AG 2003

- As of the SAP R/3 Enterprise Release, **Concurrent Employment** allows for employment of persons with multiple employment and several concurrent employment contracts with a company.
- A separate personnel number is then assigned for each contract. The multiple employments are grouped by assigning the corresponding personnel numbers to a **central person** (object type CP).
- An external **Person ID** is also determined and stored in infotype 0709 *Person ID*.
- In order to enable the concurrent employment function, you must use the logical database PNPCE. However, when you use new developments in the SAP R/3 Enterprise Release and do not use the concurrent employment function, we recommend that you use this new logical database and improved selection screen as it allows you to use more detailed settings in the report classes.

```
TABLES: PERNR.
NODES: PERAS.
INFOTYPES: 0000,      "Actions
            0002,      "Personal Data
            0006,      "Addresses
            ....

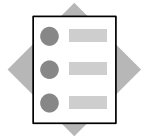
GET PERAS.
  PROVIDE * FROM P0002
    BETWEEN PN-BEGDA AND PN-ENDDA.
  WRITE...

ENDPROVIDE.
```

© SAP AG 2003

- In the report, the structure PERNR must be declared by the TABLES statement. However, further use of the structure PERNR is restricted. For example, the event GET PERNR is not permitted. Use **GET PERAS** instead. Except for the component PERNR-PERNR, all other components of the structure PERNR are no longer populated and have initial values. Therefore, programming on these values (with the exception of PERNR-PERNR) is not permitted.
- The events GET GROUP and GET PERSON are also available. When these events occur, you must declare them using NODES. If these events do not occur, it is assumed that the concurrent employment function is not required. The report then runs in PNP mode.
- Only a few companies are currently using the concurrent employment function, and further details are therefore not provided here. For more information on using the concurrent employment mode, refer to the documentation for the program SAPDBPNPCE.





**At the conclusion of this topic, you will be able to:**

- **Set up the selection screen of the logical database to suit customer-specific needs**

Report category **00000350** **HR350**

☐ For the logical database PNPCE

☐ Data selection period = Person selection period

☐ Matchcode allowed    ☐ Sort allowed    ☐ Org.str. allowed

Data select. period  
☐ No entry  
☐ Key date  
☒ Interval

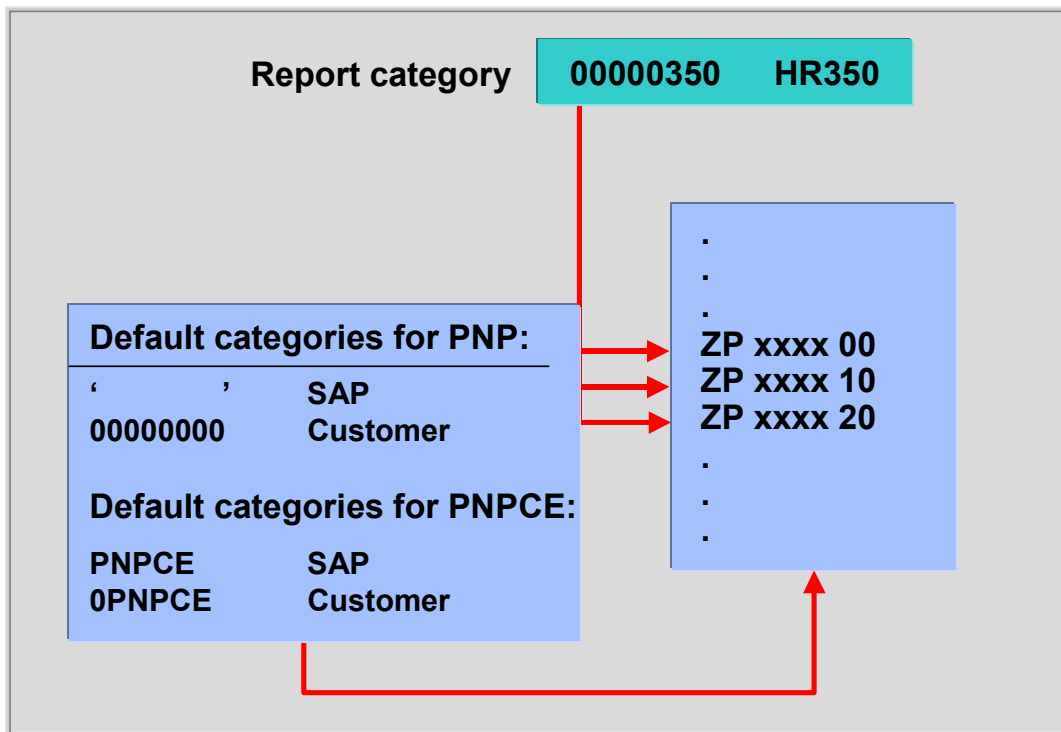
Person selection period  
☐ No entry  
☐ Key date  
☒ Interval

Payroll area/period/year  
☒ No entry  
☐ Entry possible  
☐ Mandatory field

Type **SAP**  
 Name **PNP\_DEFAULT**

© SAP AG 2003

- Think about which report categories you need. The customer-specific report categories must start with a number. Make a new entry.
- Select the field *Data selection period = Person selection period* if the person selection period is to adopt the value of the data selection period and is not to be ready for input.
- Determine whether sorting, selection using matchcodes, and access via the organizational structure are permitted.
- Decide whether the period determination indicator and the fields for the input of the data and person selection period should be displayed on the screen. Also specify whether it is possible to enter the payroll period.
- Choose a selection view for free delimitations. You can use either standard or customer-specific selection views. If no free delimitation can be selected on the selection screen, enter the view PNP\_NO\_FREE\_SEL.
- You can select the required selection fields after double-clicking *Permitted Selection Criteria* in the *Dialog Structure* group box. Select the fields that you want to appear on the first page of the selection screen. Fields that are not selected are displayed in an additional window when you choose *Further Selections*.



© SAP AG 2003

- In the standard SAP system, a report class is assigned to the reports. If you would like to override this class, carry out the steps described here. Making an assignment to report classes for standard reports using SE38 constitutes a system modification.
- For reports that you have developed yourself, you can make the assignment to report classes either using the Customizing activity described here (client-specific) or directly using the report attributes in SE38 (for all clients).
- If a report uses the logical database PNP, you can access the HR report category maintenance functions from the *Program Attributes* screen by choosing *Edit -> HR Report Category*. The report category currently assigned to the report is displayed in the dialog box. If no report category is assigned to this report in Customizing, the report category assigned by SAP is displayed.
- You can also use the SAP report categories for your customer reports. For example: You have created your own report ZPCTEST1, and you want the selection screen to correspond to the selection screen used in the evaluation reports for payroll that have the report category '\_\_M00001'. To do this, assign the report category '\_\_M00001' to your report.

1.

Create Selection View

Origin of view

CUS

☐ For logical database

☒ For any tables

Name of view

PNP\_VIEW\_1

2.

Tables

PA0002

PA0006

Other tables

© SAP AG 2003

- **Free delimitation** is one option you can use to add selection criteria to the selection screen. When you use this option, the additional selections are made directly by the logical database and performance is therefore significantly better than if you restrict the employees to be selected in the report itself.
- Call the **Object Navigator** (SE80) and from the *Workbench* menu, choose *Edit object*. Under *More...*, choose *Selection view* and choose *Create*.
- Enter a name for your view on the screen that appears. Select a name with the prefix **PNP**.
- On the next screen, enter the names of the database tables whose fields you want to use for the selection view.

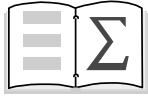
Functional groups		Tables/Nodes	
01	Personal data	PA0002	HR Master Record Infotype 0002
02	Address data	PA0006	HR Master Record Infotype 0006

Table/Node Fields		
02	STRAS	Street and house number
02	ORT01	City
02	ORT02	District
02	PSTLZ	Postal code
02	LAND1	Country code
02	TELNR	Telephone number

© SAP AG 2003

- Within a selection view, fields are selected and grouped together as **functional groups**. All fields assigned to a functional group are included in the view. By grouping the fields into functional groups, you can keep fields with similar contents together. This enables you to use the view to search for specific information.
- A functional group is defined by assigning a functional group identifier and a corresponding text. The function group identifier can consist of any two characters and is only relevant for maintaining the selection view.
- From the *Table/Node* window, select the table whose fields you want to assign to one of your defined functional groups. The system lists the fields in the *Fields in Table/Node* screen from where you can assign the required fields to a functional group.



**You are now able to:**

- **Explain the functions and data structures of the logical database PNP**
- **Process period-based data in a report**
- **Use the screening criteria to select data and persons**
- **Set up the standard selection screen**
- **Create a selection view for free delimitations**
- **Use the logical database PNPCE in PNP mode**



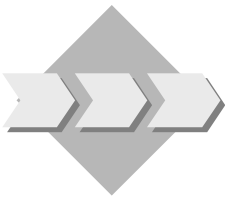
## Unit: The Logical Databases PNP/PNPCE

### Topic: Creating a List of Employees



At the end of this exercise, you will be able to:

- Process infotype records using **PROVIDE...ENDPROVIDE**



The personnel department in your company requires a list of employees specifying language and date of birth.

1-1 Create a report which generates a list of employees with the following information:

- Personnel number
- Name
- Language
- Date of birth

1-1-1 In the selection screen for the logical database, set the period determination indicator 'Today' as the default value.

1-1-2 Enable the selection of employee according to language using **SELECT-OPTIONS**.

1-1-3 Take the formatted names from the field **PERNR-ENAME**.

1-1-4 Separate the resulting field contents in your list by a vertical line (**SY-VLINE**).



Create your report using the name **ZPG##Enr**.

**##** = Group number

**nr** = Exercise number (start with 01)



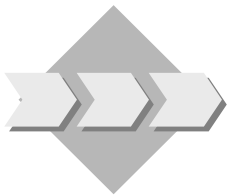
## Unit: The Logical Databases PNP/PNPCE

### Topic: Creating a Customer Report Class



At the end of this exercise, you will be able to:

- Create customer report classes using the IMG
- Assign report classes to customer reports.



The personnel department requires that the selection screens for the customer reports only display the selection fields that are required for the corresponding reports.

- 1-1 Use the IMG to create a customer report class 000000## (where nn = group number) for the logical database PNP. Sorting and matchcodes are permitted and the start and end dates are permitted entries for both the data selection period and the person selection period.

Select the following fields for the first page of the SELECT-OPTIONS:

- Personnel number
- Employment status
- Company code
- Personnel subareas
- Employee group
- Employee subgroup
- Cost center

- 1-2 Assign the new report class to your report.





## Unit: The Logical Databases PNP/PNPCE

### Topic: Creating a List of Employees

```
REPORT zpsol010.
*-- Declaration
TABLES: pernr.
INFOTYPES: 0002.                  "Personal Data
SELECT-OPTIONS: language FOR p0002-sprsl.

*-- Selection screen
INITIALIZATION.
    pnptimed = 'D'.

*-- Processing
GET pernr.
    PROVIDE * FROM p0002 BETWEEN pn-begda AND pn-enddda.
    CHECK language.
    WRITE: / p0002-pernr,
            sy-vline,
            pernr-ename,
            sy-vline,
            p0002-sprsl,
            sy-vline,
            p0002-gbdat.
ENDPROVIDE.
```



## Unit: The Logical Databases PNP/PNPCE

### Topic: Creating a Customer Report Class

#### 1-1 Create customer report class 000000##:

***IMG → Personnel Management → Human Resources Information System → Reporting → Adjusting the Standard Selection Screen → Create Report Categories***

Select *New Entries* and enter the number of your report class and the name. Deactivate the *For Logical Database PNPCE* field.

Select the *Matchcode allowed* and *Sort allowed* indicators and allow the entry of a date interval for the data and person selection period.

Save your entries and create a transport request. Return to the overview screen.

On the overview screen, select your new report class and select *Allowable Selection Criteria* in the dialog structure. Choose *New Entries*. Use the possible entries help (F4) to select the specified selection fields and select the field for output on the first page. Save your entries.

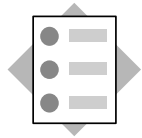
#### 1-2 To assign the customer report category to your own reports, select:

***IMG → Personnel Management → Human Resources Information System → Reporting → Adjusting the Standard Selection Screen → Assign report categories***

Choose *New Entries* and enter your program name and new report class. Save your entries.

## **Contents:**

- **Joining infotypes**
- **Field projection**
- **Combining join and projection**



**At the conclusion of this unit, you will be able to:**

- **Process two or more infotypes in one  
PROVIDE - ENDPROVIDE loop**
- **Determine that only changes to specific fields are  
taken into account when an infotype is processed**
- **Combine join and projection**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration

Data Structures in Personnel Planning

The Logical Databases PNP/PNPCE

The Logical Database PCH



**Join and Projection**

Customer-Specific Infotypes

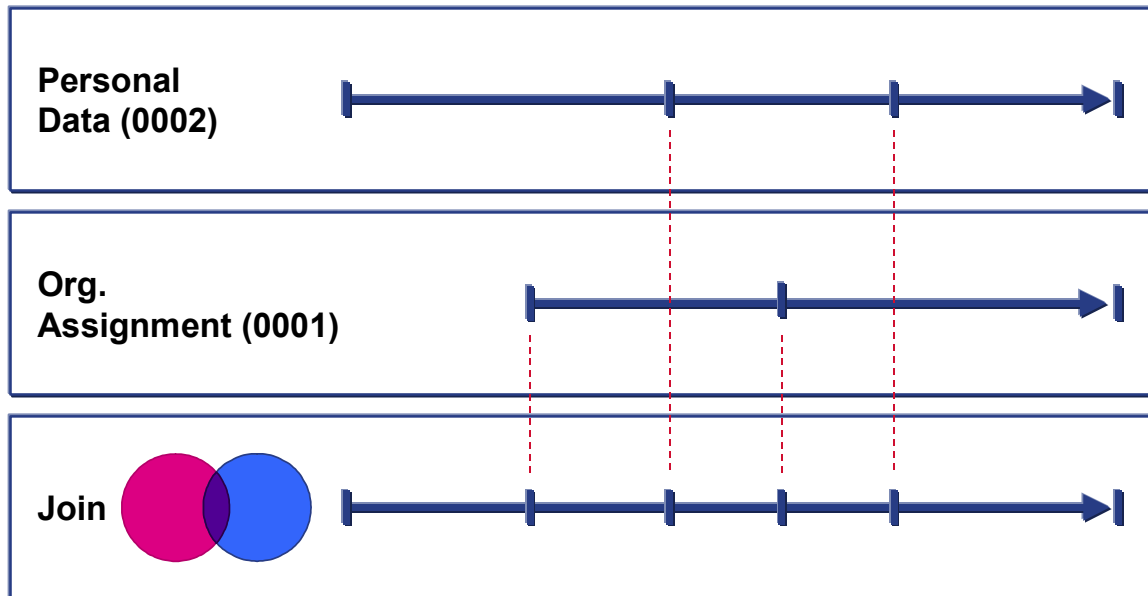
Specific Commands

HR ABAP Features

Repetitive Structures and List Display



- **The Personnel Department of your company requires additional reports that combine data from different infotypes.**



© SAP AG 2003

- Evaluations can either be coded for specific infotypes, which means that each infotype is processed in its own PROVIDE loop, or for all infotypes, which means that a single PROVIDE loop retrieves and processes data from two or more infotypes at the same time. The infotypes to be processed are listed as the data source.
- All HR data changes over time. For this reason, time-dependencies must be taken into account when infotype data is linked by a join. HR data is not infinitely valid. It is valid for specific periods only.
- Therefore, the result of a join is one or more validity intervals in which the data of both infotypes is valid.
- If an evaluation is run for key dates, the currently valid data of both infotypes is retrieved in a data record.
- In principle, a join is a logical database operation performed on the time axis. Based on the validity periods of the infotypes to be linked, new periods are created with the data that is valid. In other words, new infotype records are created. This enables you to see the time-based interrelationships between the infotypes in question.

```
TABLES: PERNR.  
INFOTYPES: 0001,      "Organizational Assignment  
            0002,      "Personal Data  
            0006,      "Addresses  
            ....  
  
GET PERNR.  
  PROVIDE * FROM P0001  
          * FROM P0002  
          BETWEEN PN-BEGDA AND PN-ENDDA.  
  WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- Any number of infotypes can be linked by a join.
- If more than one record is valid for an infotype at the same time or in the same period (time constraint 3), the data that is valid in the partial interval covered by the join is not meaningful. The resulting restriction for the join is that the data of the infotypes in question must remain unique at all times. This means that joins are only possible for infotypes with **time constraint 1 or 2**.
- The logical validity is stored in the infotype's BEGDA and ENDDA fields.
- Note that changing data in any one of the infotypes linked by a join causes a split in the selection period.

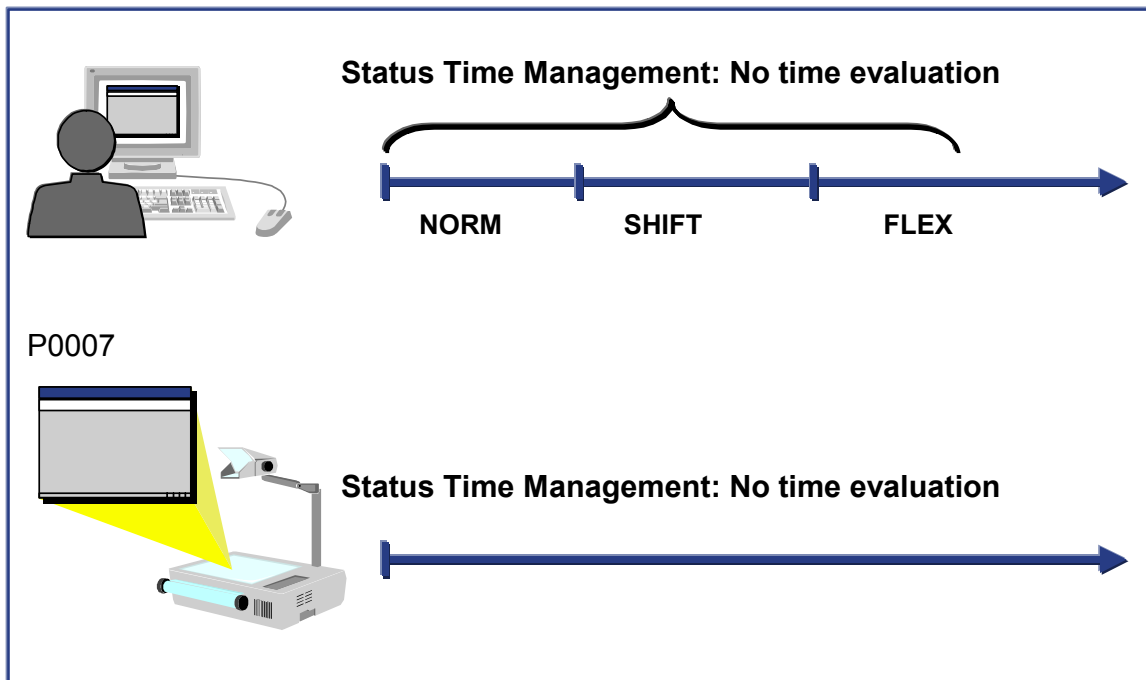


```
TABLES: PERNR.
INFOTYPES: 0001,      "Organizational Assignment
            0002,      "Personal Data
            0006,      "Addresses
            ....

GET PERNR.
  PROVIDE * FROM P0002
          * FROM P0006
            BETWEEN PN-BEGDA AND PN-ENDDA
  WHERE P0006-SUBTY = '1'.
  IF P0006_VALID = 'X'.
    WRITE...
  ENDIF.
ENDPROVIDE.
```

© SAP AG 2003

- If infotypes linked by a join have subtypes, processing must be restricted to one subtype using a WHERE condition, in which a subtype is queried.
- In this example, the first partial interval only contains personal data. The record is not meaningful because the join's task of retrieving data from all of the infotypes in question has not been performed.
- Using the variable Pnnnn\_VALID, the system recognizes that one partial interval only contains incomplete data.
- When the report is run, this variable is created for each Pnnnn infotype included in a join.
- If a partial interval for infotype Pnnnn contains data, its Pnnnn\_VALID variable is filled with X.



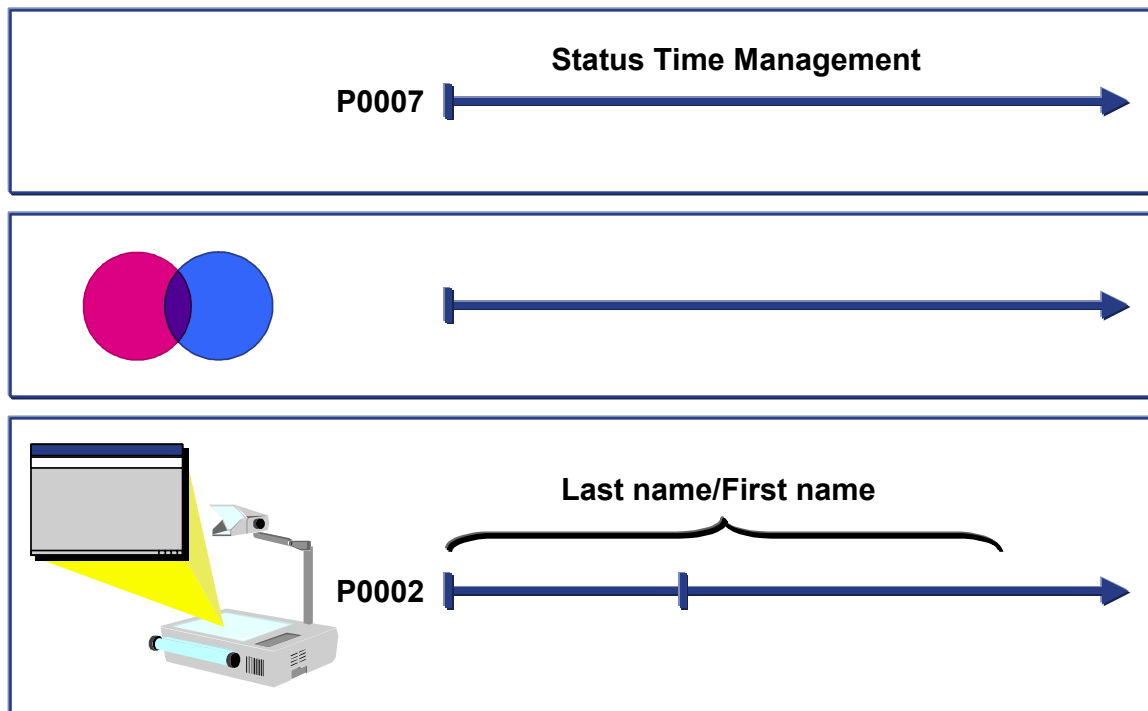
© SAP AG 2003

- Selecting one or more infotype fields for processing is called **projection**.
- Using projection for specific fields enables you to determine that only the contents of these fields, and time-based changes to the contents, are relevant to an evaluation.
- Like joins, a projection is a logical data view concentrated on one or more fields within an infotype.
- Projection is also an operation performed on the time axis. New validity periods are created when the contents of one of the projection fields change. The values stored in fields that are not included in the projection are in an undefined condition, which means they are no longer relevant to output.
- If the contents of a field included in the projection remain unchanged in several data records, the validity periods of these data records are combined.
- The process of combining data records during projection is called **contraction**.

```
TABLES: PERNR.  
INFOTYPES: 0002,      "Personal Data  
            0007,      "Planned Working Time  
            ....  
  
GET PERNR.  
  PROVIDE ZTERF FROM P0007  
            BETWEEN PN-BEGDA AND PN-ENDDA.  
  WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- The \* in the PROVIDE statement, which stands for all of the fields in the infotype, is replaced by the five-digit field name of the infotype included in the projection.
- The \* in a simple PROVIDE means that all of the fields in the infotype are included in the projection.
- Join and projection can be combined in a PROVIDE statement.
- Infotype records are not allowed to overlap for projection. Data must be unique, which means that projection is only allowed for **time constraints 1 and 2**.



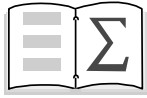
© SAP AG 2003

- These logical views of infotype data - joins and projection - can be combined.
- Data from several infotypes is read at the same time, and new partial periods are created. Since fields are only selected from infotypes if they are relevant to an evaluation, partial periods are combined.
- In this way, validity periods with related contents are combined, and data is retrieved for an evaluation from relevant infotype fields and for an exact period.

```
TABLES: PERNR.  
INFOTYPES: 0002,      "Personal Data  
            0007,      "Planned Working Time  
            ....  
  
GET PERNR.  
  PROVIDE  NACHN  
           VORNA FROM P0002  
           ZTERF FROM P0007  
           BETWEEN PN-BEGDA AND PN-ENDDA.  
  
  WRITE...  
  
ENDPROVIDE.
```

© SAP AG 2003

- Join and projection can be combined in a PROVIDE statement.



**You are now able to:**

- **Process two or more infotypes in one PROVIDE - ENDPROVIDE loop**
- **Determine that only changes to specific fields are taken into account when an infotype is processed**
- **Combine join and projection**



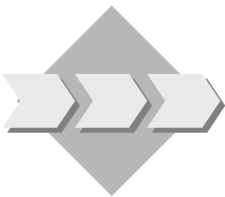
## Unit: Join and Projection

### Topic: Joining Infotypes



At the end of this exercise, you will be able to:

- Process the *Personal Data* infotype (0002) and the *Addresses* infotype (0006) using a join.
- Make the processing of data dependent on the content of the variable Pnnnn\_VALID.



The personnel department of your company requires an overview of all previous and present addresses of employees.

- 1-1 Create a report that documents the period in which the employee lived at which address and under which name. The created list should contain the following information:
- Personnel number
  - Start date
  - End date
  - Name
  - Place of residence
- 1-1-1 Only process the records from infotype 0006 which have subtype 1.
- 1-1-2 Only print data for the periods for which address information is available.
- 1-1-3 Use a variable with the characteristics of field **P0001-ENAME**, so that the first and last name are only separated by a space. Use the **CONCATENATE** statement.
- 1-1-4 Enter a width of 100 characters for the list (addition **LINE-SIZE** for keyword **REPORT**).  
Limit the length of the name to 30, and the length of the city to 20 characters.  
Separate the resulting field contents in your list by a vertical line (**SY-VLINE**).



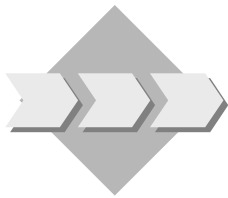
## Unit: Join and Projection

### Topic: Projection to Infotype Fields



At the end of this exercise, you will be able to:

- Project to particular fields by joining two infotypes



Your company's Board of Directors requires a list of employees with the values of the *Status of Time Evaluation* field in the *Planned Working Time* infotype.

- 1-1 Create a report that documents the values of the *Status of Time Evaluation of Employees* field. The list should contain the following information:
  - Personnel number
  - Name
  - Status of time recording
  - Start date
  - End date
- 1-1-1 Project to the fields **P0002-NACHN** and **P0002-VORNA** for infotype 0002. Project to field **P0007-ZTERF** for infotype 0007.
- 1-1-2 Only print data for the periods for which address information is available.
- 1-1-3 The texts for the keys in the *Status of Time Recording* field are contained in the table T555V. Before reading this table, check that the table entry to be read for the corresponding key in question is already in the work area. To do this, use the **CHECK** statement.
- 1-1-4 Enter a width of 100 characters for the list (addition **LINE-SIZE** for keyword **REPORT**).





## Unit: Join and Projection

### Topic: Joining Infotypes

```
REPORT zpsol020 LINE-SIZE 100.
*-- declaration
TABLES: pernr.
INFOTYPES: 0002,           "Personal Data
            0006.           "Address
DATA: name LIKE pernr-ename.

*-- Processing
GET pernr.
  PROVIDE * FROM p0002
            * FROM p0006 BETWEEN pn-begda AND pn-enddda
            WHERE p0006-subty = '1'.
  IF p0006_valid EQ 'X'.
    CONCATENATE p0002-nachn p0002-vorna INTO name
    SEPARATED BY SPACE.
    PERFORM print_data.
  ENDIF.
ENDPROVIDE.

*-----*
*      FORM PRINT_DATA
*-----*
FORM print_data.
  WRITE: / p0002-pernr NO-ZERO,
          sy-vline,
          p0002-begda,
          sy-vline,
          p0002-enddda,
          sy-vline,
          (30) name,
          sy-vline,
          (20) p0006-ort01.
ENDFORM.
```



## Unit: Join and Projection

### Topic: Projection to Infotype Fields

```

REPORT zpsol030 LINE-SIZE 100.
*-- Declaration
TABLES: pernr,
        t555v.                                "Job Titles
INFOTYPES: 0002,                              "Personal Data
            0007.                              "Planned Working Time
DATA: name LIKE pernr-ename.

*-- Processing
GET pernr.
  PROVIDE nachn vorna FROM p0002
        zterf          FROM p0007 BETWEEN pn-begda AND pn-endda.
  IF p0007_valid EQ 'X'.
    CONCATENATE p0002-nachn p0002-vorna INTO name
    INTO name SEPARATED BY space.
    PERFORM re555v USING p0007-zterf.
    WRITE: / pernr-pernr,
            name,
            (20) t555v-ztext,
            p0007-begda,
            p0007-endda.
  ENDIF.
ENDPROVIDE.

*-----*
*      FORM RE555v                                *
*-----*
*      Read Time Management Status                *
*-----*
FORM re555v USING value(tm_status).
  CHECK sy-langu NE t555v-sprsl OR
        tm_status NE t555v-zterf.

  SELECT SINGLE * FROM t555v WHERE sprsl EQ sy-langu
                                AND zterf EQ tm_status.

  IF sy-subrc NE 0.
    MOVE space TO t555v.
  ENDIF.
ENDFORM.

```

### **Contents:**

- **Macros**
- **Function modules**

© SAP AG 2003



**At the conclusion of this unit, you will be able to:**

- **Use macros**
- **Use function modules**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration

Data Structures in Personnel Planning

The Logical Databases PNP/PNPCE

The Logical Database PCH

Join and Projection

Customer-Specific Infotypes

 **Specific Commands**

HR ABAP Features

Repetitive Structures and List Display



- **The Executive Board of your company requires lists that display the data valid for an employee in a specified data selection period.**



**At the conclusion of this topic, you will be able to:**

- **Use macros for Personnel Administration logical databases**

© SAP AG 2003

- Like subroutines and function modules, macro modules can be used in HR to modularize source code.
- For documentation on key macros, see the section on programming aids for the logical databases PNP and PAP in SAP Library.

```
TABLES: PERNR.  
INFOTYPES: 0001,      "Organizational Assignment  
           0002,      "Personal Data  
           0006,      "Addresses  
           ....
```

```
GET PERNR.
```

```
RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGDA PN-ENDDA.
```

```
WRITE...
```

```
* * * Include program DBPNPMAC.
```

```
DEFINE RP_PROVIDE_FROM_LAST.
```

```
PNP-SW-FOUND = '0'.
```

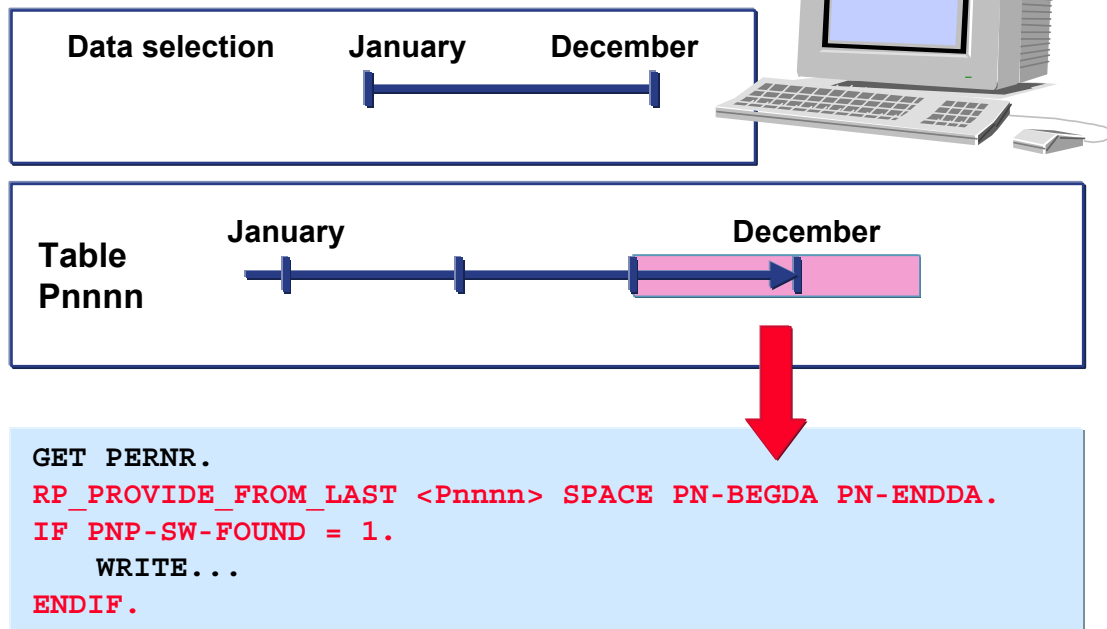
```
. . .
```

```
END-OF-DEFINITION.
```

© SAP AG 2003

- Like subroutines and function modules, macro modules can be used to modularize programs. Macro modules are frequently used in HR.
- These macros are defined in the program SAPDBPNP (include DBPNPMAC) with the keyword DEFINE. They can be used in any program that uses the logical database PNP.
- If you want to use these macros in reports that do not use the logical database PNP, you must include program DBPNPMAC with the keyword INCLUDE.
- You can also define your own macros. In accordance with the naming convention, the first two letters stand for the application.
- Some macros are also stored in the table Macros in ABAP Programs (TRMAC)





© SAP AG 2003

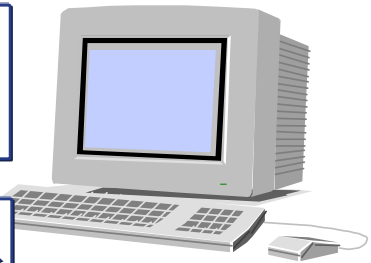
- The `RP_PROVIDE_FROM_LAST` macro retrieves the last valid data record in the data selection period.
- The parameters for `RP_PROVIDE_FROM_LAST` are: Infotype, subtype, start date, and end date. If you do not want to specify a particular subtype, enter `SPACE`.
- As well as processing the last valid data record in the data selection period, you can also process the first valid data record using the `RP_PROVIDE_FROM_FRST` macro.
- The `PNP-SW-FOUND` macro return code has the value 1 if there is a corresponding entry in the infotype table in the specified period. It has the value 0 if no entry was found.

**Data Selection**

**May      December**



**PAnnnn**



**START-OF-SELECTION.**

**RP\_SET\_DATA\_INTERVAL '<Pnnnn>' PN-BEGDA PN-ENDDA.**

**GET PERNR.**



**Table  
Pnnnn**



© SAP AG 2003

- When the START-OF-SELECTION event occurs, the RP\_SET\_DATA\_INTERVAL macro ensures that the only records imported from database table PAnnnn to internal table Pnnnn are those which are valid in the data selection period.
- If the selection specified is to apply to all infotypes, the infotype name in quotation marks must be replaced by **ALL**.

```
TABLES: PERNR, T001P.
INFOTYPES: 0001,
        ....
```

## Personnel Area/Subarea Table

PersArea	Subarea	Groupings ...
...	...	...
CABB	0001	01 01
...	...	...

```
GET PERNR.
RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGDA PN-ENDDA.

RP-READ-T001P P0001-WERKS P0001-BTRTL SPACE.
```

© SAP AG 2003

- It is often necessary to read the groupings stored in the Personnel Area/Subarea table because they are required as keys for other tables. This must take place separately for each personnel number to be processed.
- Each employee's assignment to a personnel area and subarea is stored in infotype 0001.
- The parameters of the macro are:
  - Personnel area
  - Personnel subarea
- If no entry is found, the system reacts as follows:
  - X = Termination of report with the error message "No entry found in table ..."
  - SPACE = SY-SUBRC is set to 4
- Alternatively, if you only want to determine the country indicator, you can use the function module **HR\_COUNTRYGROUPING\_GET**.

```

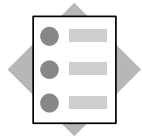
INFOTYPES:  0001,
            0002,
            ...
            2005 MODE N.

GET PERNR.
  RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGDA PN-ENDDA.
  . . . .
  RP_READ_ALL_TIME_ITY PN-BEGDA PN-ENDDA.
  . . . .

  LOOP AT P2005.
    WRITE...
  ENDLOOP.
    
```

© SAP AG 2003

- Due to the large amount of data in HR, the infotypes 2000 – 2999 should not be read when GET PERNR occurs. Therefore, these infotypes are declared with the enhancement **MODE N**.
- As a result, the infotype tables under GET PERNR are not filled. The time infotype tables are filled subsequently using the macro **RP\_READ\_ALL\_TIME\_ITY**, but only for the time interval specified by PN-BEGDA and PN-ENDDA.

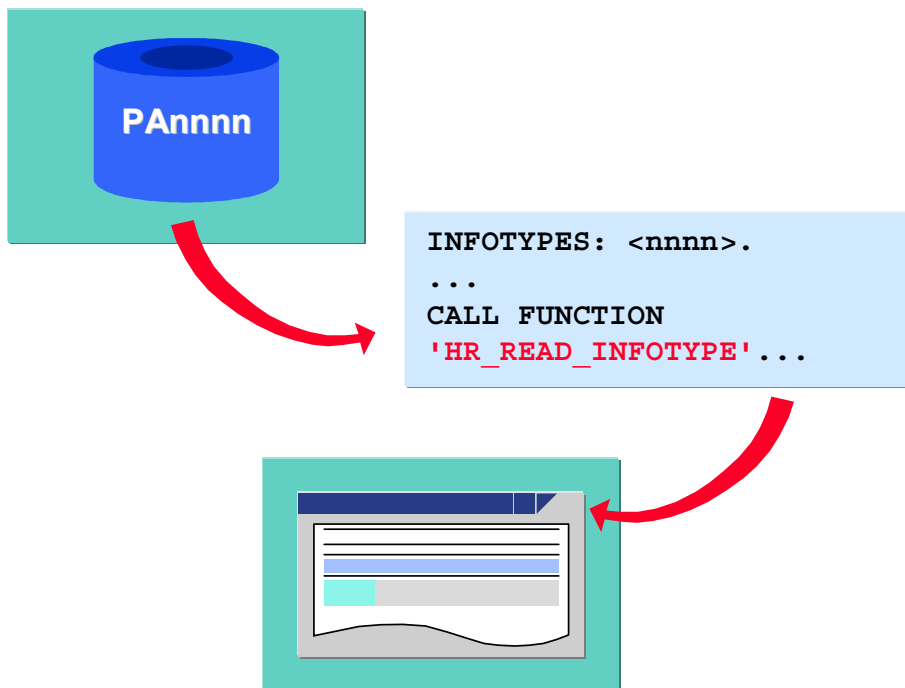


**At the conclusion of this topic, you will be able to:**

- **Search for function modules in HR, to use them in your own programs**

© SAP AG 2003

- You can use the standard function modules to help you process large quantities of data. Make sure that you check whether allowed entries are not already defined for specific functions in standard function modules.
- Most function modules in HR start with **"RP"** ,**"HR"** and **"BAPI"**. In transaction SE37, select the menu *Utilities -> Find*. For example, if you are looking for a function module that determines an employee's hiring date, search for the entries the **"RP\*"** and **"HR\*"** in the selection criterion *Function module* and **"\*Entry\*"** in the *Short description* field.



© SAP AG 2003

- You can also read infotype records for a particular personnel number without using the logical database.
- To do this, use the function module `HR_READ_INFOTYPE`. However, you must ensure that the internal table for the required infotype is declared with the `INFOTYPE` statement.

```

INFOTYPES: 0002.
DATA: return LIKE SY-SUBRC.

CALL FUNCTION 'HR_READ_INFOTYPE'
  EXPORTING
    . . .
    PERNR          = <person>
    INFY           = '0002'
    BEGDA          = <begdat>
    ENDDA          = <enddat>

  IMPORTING
    SUBRC          = return

  TABLES
    INFY_TAB       = P0002

  EXCEPTIONS
    INFY_NOT_FOUND = 1
    OTHERS         = 2.
    
```

© SAP AG 2003

- The function module reads the HR infotype records for a person (employee or applicant) in accordance with the specified selection criteria. Values are returned in an internal table, the structure of which corresponds to the appropriate infotype table. In the calling program, such tables can be declared with the INFOTYPES statement, for example. An infotype record is selected if its validity period overlaps with the specified period.
- The function module carries out an authorization check.
- The return code can have the following values:
  - 0: The return table contains all the records requested
  - 4: The return table contains records, but is incomplete due to missing authorization
  - 8: The return table is empty as no records were found for the criteria specified
  - 12: The return table is empty due to missing authorization
- **Note:**  
 You should not use this function module in reports that use the logical database PNP. If you want to read an infotype separately in one of these reports, you can call the subroutine **READ-INFOTYPE** directly in the database program **SAPDBPNP** (PERFORM READ-INFOTYPE(SAPDBPNP)). . .). In this case, you declare the infotypes in the statement INFOTYPES with enhancement **MODE N**.

```
INFOTYPES: 0001.
DATA: hire_date LIKE P0016-EINDT.

CALL FUNCTION 'RP_GET_HIRE_DATE'
  EXPORTING
    PERSNR          = P0001-PERNR
    CHECK_INFOTYPES = '0000'
  *   DATUMSART      = '01'
  *   STATUS2        = '3'
  *   P0016_OPTIONEN = ' '
  IMPORTING
    HIREDATE        = hire_date
  EXCEPTIONS
    OTHERS           = 1.
```

© SAP AG 2003

- You can use this function module to determine an employee's entry date. The following infotypes can be used to determine this date:
  - P0000 Measures
  - P0001 Organizational assignment
  - P0016 Contract components
  - P0041 Date specifications
- The transfer parameter **CHECK\_INFOTYPES** is used to determine which of these infotypes are taken into account. The employment status (STATUS2 parameter) can also be specified for infotype P0000.
- If the infotype P0016 is used, you can decide whether the fields P0016-EINDT and/or P0016-KONDT are used to determine dates. If this transfer parameter is SPACE, both fields are taken into account. Specify the technical field name in the transfer parameter for only one field.
- For infotype P0041, the corresponding date type can be specified in the transfer parameter DATUMSART. The date type '01' is usually used for the technical entry date.



```
DATA: return_struct TYPE BAPIRETURN1,
. . .

CALL FUNCTION 'BAPI_EMPLOYEE_ENQUEUE'
  EXPORTING
    NUMBER          = PERNR-PERNR
  IMPORTING
    RETURN          = return_struct.

IF NOT return_struct IS INITIAL.
  WRITE: / return_struct-TYPE,...
ENDIF.
. . .
* Update Infotype Records
. . .
CALL FUNCTION 'BAPI_EMPLOYEE_DEQUEUE'
  EXPORTING
    NUMBER          = PERNR-PERNR
  IMPORTING
    RETURN          = return_struct.
```

© SAP AG 2003

- Before you change employee data, you must lock the personnel number. You can do this using the function module **BAPI\_EMPLOYEE\_ENQUEUE**.
- Messages are returned in the **RETURN** parameter. If errors occur, this structure contains the following information, for example:
  - Message type (*TYPE* field)
  - Message text (*MESSAGE* field)
 If no errors occur, the structure is blank.
- If an employee's data has been locked, only the user who has locked the data can access the data records linked to the employee. Access is denied to other users. You must therefore unlock the data after it has been changed. You can do this using the function module **BAPI\_EMPLOYEE\_DEQUEUE**.

```

DATA:   return_struct TYPE bapireturn1,
        record_key    TYPE bapipakey.
. . .
GET PERNR.
  LOOP AT P0002 WHERE. . .

      CALL FUNCTION 'HR_INFOTYPE_OPERATION'
        EXPORTING
          INFTY          = '0002'
          NUMBER         = P0002-PERNR
          LOCKINDICATOR  = P0002-SPRPS
          VALIDITYEND    = P0002-ENDDA
          VALIDITYBEGIN  = P0002-BEGDA
          RECORDNUMBER   = P0002-SEQNR
          RECORD         = P0002
          OPERATION      = 'MOD'
          DIALOG_MODE    = '1'
        IMPORTING
          RETURN         = return_struct
          KEY            = record_key.

  ENDLOOP.

```

© SAP AG 2003

- You can use this function module to maintain employee and applicant master data. You can specify one data record. All validations that would normally take place if the infotypes were maintained in dialog mode with the individual maintenance screens are also carried out here. The module returns the same error messages as in dialog mode. This means the error messages for the individual maintenance screens are displayed rather than interpreted. The update is carried out by a 'Call dialog' to the module pool for the infotypes. This means that some restrictions apply to the infotypes processed in that way (see the documentation for the module).
- The following values are amongst those available for the **OPERATION** parameter: MOD (change), COPY (copy), DEL (delete), INS (insert), LIS9 (delimit).
- You can use the **DIALOG\_MODE** parameter to specify whether the action is generally run in the background or whether it only runs in the background until an error occurs, or whether the changes are generally carried out in dialog mode. Possible values:
  - '0' The changes are generally processed in the background. If an error occurs (with an E or A message), the entire step is terminated and the module returns the corresponding error message in the 'RETURN' structure.
  - '1' The changes are processed in the background. If an error occurs, the system switches to dialog mode so that the user can correct the entries.
  - '2' The changes are processed in dialog mode.

```
PARAMETERS:natio_o LIKE P0002-NATIO DEFAULT 'DE',
            natio_n LIKE P0002-NATIO DEFAULT 'D'.

* Update Infotype-records

UPDATE PA0002
      SET NATIO      = natio_n
      WHERE PERNR    = P0002-PERNR
      AND NATIO      = natio_o.

IF SY-SUBRC = 0.
  WRITE: 'Modified records', SY-DBCNT.
ENDIF.
```

© SAP AG 2003

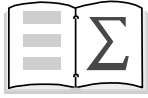
- If you cannot use the function module HR\_INFOTYPE\_OPERATION, you can use the UPDATE statement. Note that the system does not check whether the new field contents are correct when a direct database update takes place.
- In this example, the field P0002-NATIO (nationality) is updated directly in DB table PA0002. The field contents are changed from "DE" to "D". The incorrect value "D" in the field PA0002-NATION then triggers an error message when the infotype record is changed or copied in dialog mode.
- The SY-DBCNT system field contains the number of changed records.
- Authorization checks are not supported by the UPDATE statement and should be carried out at the program level.

```
CALL FUNCTION 'HR_CHECK_AUTHORITY_INFTY'
  EXPORTING
    TCLAS      = 'A'
    PERNR      = <person>
    INFTY      = '0006'
    SUBTY      = '*'
    BEGDA      = <begdat>
    ENDDA      = <enddat>
    LEVEL      = 'R'
  EXCEPTIONS
    NO_AUTHORIZATION = 1
    INTERNAL_ERROR   = 2
    OTHERS           = 3.

IF SY-SUBRC = 0.
  SELECT * FROM pa0006
        INTO TABLE...
ENDIF.
```

© SAP AG 2003

- Authorization checks are not supported by the SELECT statement and must therefore be executed at the program level.
- In this case, you must use the function module **HR\_CHECK\_AUTHORITY\_INFTY** to check whether the user has the required authorization for data and persons.
- If the infotype does not have any subtypes, you can use the SPACE value with the SUBTY parameter.
- SAP recommends that you use the logical database PNP as it automatically runs authorization checks.



**You are now able to:**

- **Use macros**
- **Use function modules**





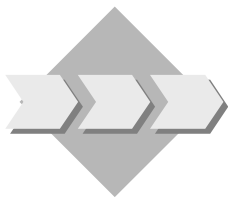
## Unit: Specific Commands

### Topic: Macros



At the end of this exercise, you will be able to:

- Use the `RP_SET_DATA_INTERVAL` macro to read only the records valid in the evaluation period.
- Use the `RP_PROVIDE_FROM_LAST` macro to retrieve the last valid record in the data selection period.
- Use the `RP-READ-T001P` macro to read the Personnel Area/Personnel Subarea table.



The Executive Board of your company requires a list of employees with recurring payments and deductions.

- 1-1 Create a report that will generate a list of the recurring payments and deductions for each employee.

The list should contain the following information for each employee:

- Personnel number
- Name
- Personnel area
- Personnel subarea text

In the following lines, print:

- Wage type
- Wage type text
- Amount
- Start date
- End date

- 1-1-1 Use the **RP\_SET\_DATA\_INTERVAL** macro to specify that only records of infotype 0014 are read for the evaluation period. Only employees who have recurring payments or deductions in this period should appear on the list. If employees do not have a record for infotype 0014, the internal table P0014 is empty when the event GET PERNR occurs.
- 1-1-2 To place the last infotype 0001 record valid in the data selection period in the header of the internal table P0001, use the macro **RP\_PROVIDE\_FROM\_LAST**.
- 1-1-3 Use the **RP-READ-T001P** macro to read the table T001P (Personnel Area/Subarea). The country indicator is assigned to each personnel area/subarea in the field **T001P-MOLGA**. The country indicator is part of the key for table T512T (Wage Type Texts). Use the personnel area text in field **T001P-BTEXT**.
- 1-1-4 Read the corresponding wage type texts for the wage types for infotype 0014 from table T512T. Before reading table T512T, check that the table entry to be read is already in the work area. To do this, use the **CHECK** statement. Use a maximum of 20 characters for the name.





## Unit: Specific Commands

### Topic: Macros

```
REPORT zpsol040.
*-- Declaration
TABLES: pernr,
        t512t,           "Wage type text
        t001p.           "Plant section
INFOTYPES: 0001,        "Organ. assignment
           0014.        "Recur.Payments/Deds.

Start-of-selection.
  rp_set_data_interval 'P0014' pn-begda pn-enddda.

*-- Processing
GET pernr.
  CHECK lines( p0014 ) GT 0.
  rp_provide_from_last p0001 space pn-begda pn-enddda.
  rp-read-t001p p0001-werks p0001-btrtl space.
  SKIP 2.
  WRITE: / pernr-pernr,
         (20) pernr-ename,
         p0001-werks,
         t001p-btext.

  LOOP AT p0014 WHERE enddda GE pn-begda
        AND begda LE pn-enddda.
    CHECK p0014-betrg NE 0.
    PERFORM print USING
      p0014-lgart
      p0014-betrg
      p0014-begda
      p0014-enddda.

  ENDLOOP.
```

```

*-----*
*          FORM PRINT                                *
*-----*
*          Print List                                *
*-----*

```

```

FORM print USING
      value(wtype)
      value(amount)
      value(begda)
      value(endda).

```

```

PERFORM re512t USING t001p-molga wtype.
WRITE: / wtype UNDER pernr-ename,
        t512t-lgtxt,
        amount,
        begda,
        endda.

```

```

ENDFORM.                                "print

```

```

*-----*
*          FORM RE512T                                *
*-----*
*          Read Wagetyp Texts                        *
*-----*

```

```

FORM re512t USING value(country_grouping)
                  value(wtype).

CHECK t512t-sprsl NE sy-langu
      OR t512t-molga NE country_grouping
      OR t512t-lgart NE wtype.

SELECT SINGLE * FROM t512t
              WHERE sprsl EQ sy-langu
              AND   molga EQ country_grouping
              AND   lgart EQ wtype.

IF sy-subrc NE 0.
  CLEAR t512t.
ENDIF.
ENDFORM.

```

## **Contents:**

- **Evaluating infotypes with repetitive structures**
- **Displaying lists with the ABAP List Viewer**



**At the conclusion of this unit, you will be able to:**

- **Evaluate infotypes with repetitive structures in loop processing**
- **Display lists using the ABAP List Viewer**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration

Data Structures in Personnel Planning

The Logical Databases PNP/PNPCE

The Logical Database PCH

Join and Projection

Customer-Specific Infotypes

Specific Commands

HR ABAP Features



**Repetitive Structures and List Display**

© SAP AG 2003



- Your company's Personnel Department would like to edit an overview of all employees with data in the *Date Specifications* infotype.

Date specifications 0041

Date type		Date
01	Technical entry date	January 01, 1994
40	First working day	January 16, 1994
—	_____	_____
—	_____	_____
—	_____	_____
—	_____	_____

Dictionary P0041

...	
DAR01	Date type
DAT01	Date per date type
DAR02	...
DAT02	...
...	

0005    Leave Entitlement

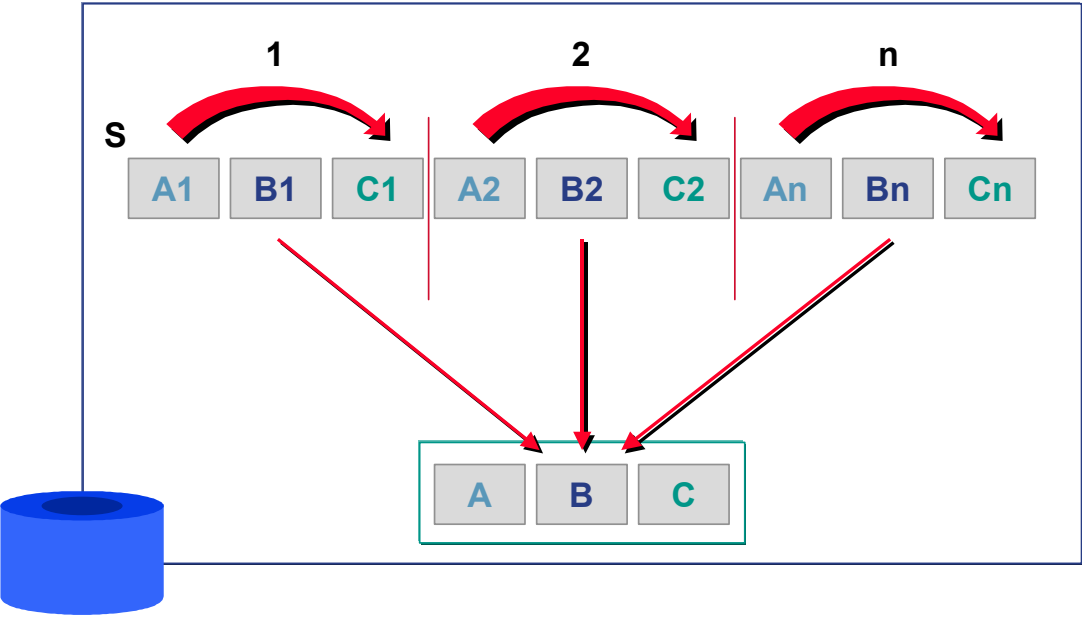
0008    Basic Pay

0025    Appraisals

0027    Cost Distribution

0041    Date Specifications

0052    Standard Wage  
         Maintenance



© SAP AG 2003

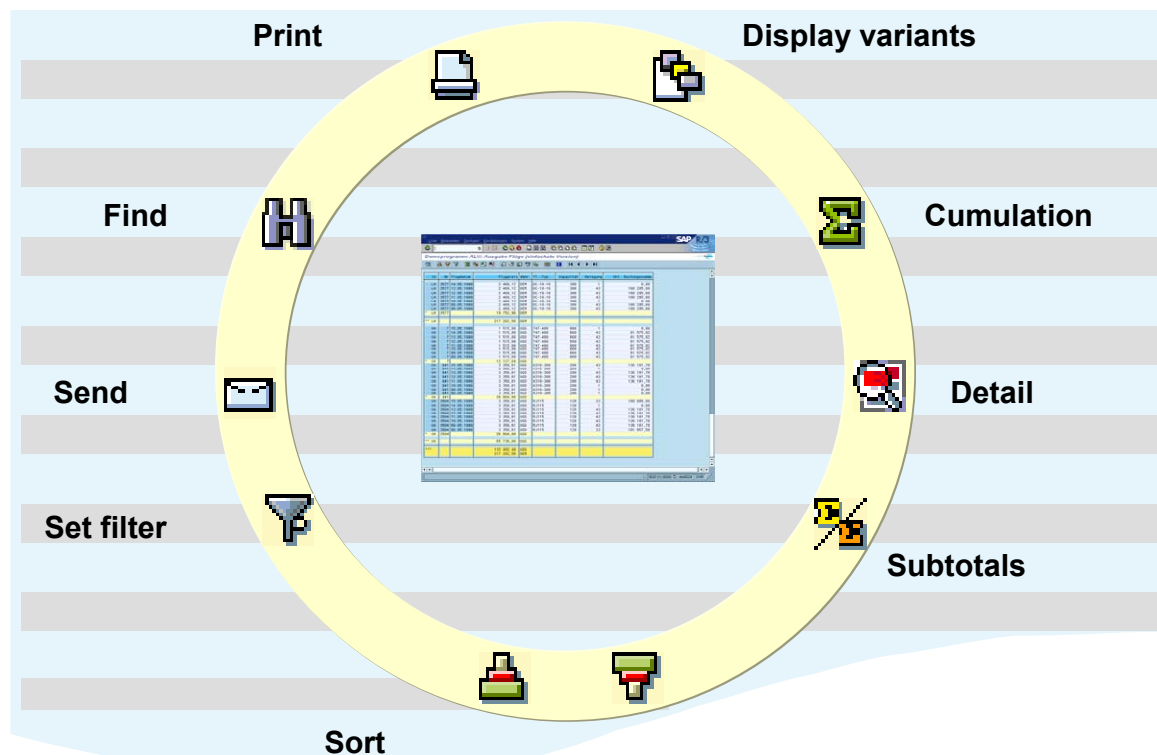


```
INFOTYPES: 0041,      "Date Specifications
            0008.      "Basic Pay
```

```
DATA:  BEGIN OF specification,
        dar LIKE P0041-DAR01,
        dat LIKE P0041-DAT01,
      END OF specification.
```

```
DATA:  BEGIN OF wagetypes,
        lga LIKE P0008-LGA01,
        bet LIKE P0008-BET01,
        anz LIKE P0008-ANZ01,
        ein LIKE P0008-EIN01,
        opk LIKE P0008-OPK01,
      END OF wagetypes.
```

```
GET PERNR.  
RP_PROVIDE_FROM_LAST P0041 SPACE PN-BEGDA PN-ENDDA.  
  
DO 12 TIMES VARYING specification  
  FROM P0041-DAR01          "Starting point  
  NEXT P0041-DAR02.         "Increment  
  IF specification-dar IS INITIAL.  
    EXIT.  
  ELSE.  
    WRITE: / specification-dar ....  
  ENDIF.  
ENDDO.
```



© SAP AG 2003

```
. . .
TYPE-POOLS: SLIS.
. . .
DATA:    alv_fieldcat TYPE slis_t_fieldcat_alv,
         alv_layout TYPE slis_layout_alv.
. . .

TYPES:   BEGIN OF data_struct,
         pernr LIKE PERNR-PERNR,
         ename LIKE P0001-ENAME,
         . . .
         dar   LIKE P0041-DAR01,
         . . .
         END OF data_struct.

DATA:    data_tab TYPE TABLE OF data_struct,
         data_tab_wa TYPE data_struct.

DATA:    stru_disvar TYPE disvariant.
```

© SAP AG 2003

```
FORM fieldcat_init USING p_fieldcat
                        TYPE slis_t_fieldcat_alv.
  DATA: ls_fieldcat TYPE slis_fieldcat_alv.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname      = 'PERNR'.
  ls_fieldcat-ref_tabname   = 'PERNR'.
  ls_fieldcat-key           = 'X'.
  APPEND ls_fieldcat TO p_fieldcat.

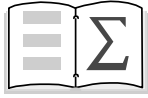
  . . .
  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname      = 'DAR'.
  ls_fieldcat-ref_fieldname  = 'DAR01'.
  ls_fieldcat-ref_tabname   = 'P0041'.
  APPEND ls_fieldcat TO p_fieldcat.

  . . .
ENDFORM.
```

```
END-OF-SELECTION.
  PERFORM fieldcat_init USING alv_fieldcat.
  alv_layout-colwidth_optimize = 'X'.
  . . .

  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
      I_CALLBACK_PROGRAM      = 'Reportname'
    . . .
      I_GRID_TITLE            = 'Title'
    . . .
      IS_LAYOUT                = alv_layout
      IT_FIELDCAT              = alv_fieldcat
      I_SAVE                   = 'A'
      IS_VARIANT               = stru_disvar
    . . .
  TABLES
    T_OUTTAB                  = data_tab
  EXCEPTIONS
    PROGRAM_ERROR             = 1
    OTHERS                    = 2.
```

© SAP AG 2003



**You are now able to:**

- **Evaluate infotypes with repetitive structures in loop processing**
- **Display lists using the ABAP List Viewer**







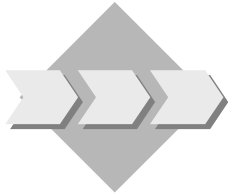
## Unit: Repetitive Structures and List Display

### Topic: Date Specifications Infotype



At the end of this exercise, you will be able to:

- Evaluate the repetitive structures in the *Date Specifications* infotype
- Display the list using the function module `REUSE_ALV_GRID_DISPLAY`



The Personnel Department requires a report that creates an overview of employees with records for the *Date Specifications* infotype. This overview can then be used for further processing.

1-1 Create a report to format the following employee data using the ABAP List Viewer:

- Personnel number
- Name
- Entry date
- Date type
- Text for date type
- Date

The report should also enable you to restrict the list to one or more date types, and only employees with records for the *Date Specifications* infotype (0041) should appear in the overview.

The user can store display variants.

1-1-1 To place the last records for infotypes 0001 and 0041 that are valid in the selection period in the header of the internal tables P0001 and P0041, use the macro **RP\_PROVIDE\_FROM\_LAST**.

For infotype 0041, query the return code PNP-SW-FOUND for the macro. If the employee has no valid record in the data selection period, the return value is 0. End further processing of the personnel numbers with the **REJECT** statement.

1-1-2 Read the texts for the date types from table **T548T**.

1-1-3 To determine the hiring date from the *Actions* infotype, use the function module **RP\_GET\_HIRE\_DATE**.

- 1-1-4 Set the layout parameters **COLWIDTH\_OPTIMIZE** and **ZEBRA** to the value “X”.
- 1-1-5 You can use the following includes:  
**ZPSOL050\_DATA\_STRUCTURES** contains the required data definitions.  
**ZPSOL050\_FILL\_FIELDCAT** contains the subroutines for filling the field catalog.



## Unit: Repetitive Structures and List Display

### Topic: Date Specifications Infotype

REPORT zpsol050.

\*-- declaration

TABLES: pernr, t548t.	"Date Types
INFOTYPES: 0001,	"Org.Assignment
0041.	"Date Specifications

```

*$*$$- Start of INCLUDE ZPSOL050_DATA_STRUCTURES          --$*$*
*&-----*
*& Include                ZPSOL050_DATA_STRUCTURES          *
*&-----*

```

TYPE-POOLS: slis.

DATA: BEGIN OF dtype,	"Work area for IT0041
dar LIKE p0041-dar01,	
dat LIKE p0041-dat01,	
END OF dtype.	

TYPES: BEGIN OF data_struct,	"Data table structure
pernr LIKE pernr-pernr,	
ename LIKE pernr-ename,	
eindt LIKE p0016-eindt,	
dar LIKE p0041-dar01,	
dtext LIKE t548t-dtext,	
dat LIKE p0041-dat01,	
END OF data_struct.	

DATA: data\_tab TYPE TABLE OF data\_struct,  
      data\_tab\_wa TYPE data\_struct.

DATA: alv\_fieldcat TYPE slis\_t\_fieldcat\_alv,  
      alv\_layout TYPE slis\_layout\_alv.

DATA: filled\_lines LIKE sy-index,  
      stru\_disvar TYPE disvariant.

```

*$*$$$--- End of INCLUDE ZPSOL050_DATA_STRUCTURES          --$*$*

```

SELECT-OPTIONS: datatype FOR p0041-dar01 DEFAULT '01'.

```

*-- Processing
GET pernr.
  CHECK lines( p0041 ) GT 0.
  rp_provide_from_last p0001 space pn-begda pn-enddda.
  rp_provide_from_last p0041 space pn-begda pn-enddda.
  IF pnp-sw-found EQ 0.
    REJECT.
  ELSE.
    DO 12 TIMES VARYING dtype FROM p0041-dar01
                                NEXT p0041-dar02.
    IF dtype-dar IS INITIAL.
      EXIT.
    ELSE.
      IF dtype-dar IN datatype.
        CLEAR data_tab_wa.
        MOVE-CORRESPONDING pernr TO data_tab_wa.
        MOVE-CORRESPONDING dtype TO data_tab_wa.
        PERFORM read_hiredate.
        PERFORM re548t USING sy-langu dtype-dar.
        MOVE t548t-dtext TO data_tab_wa-dtext.
        APPEND data_tab_wa TO data_tab.
      ENDIF.
    ENDIF.
  ENDDO.
ENDIF.

```

```

END-OF-SELECTION.
  PERFORM fieldcat_init USING alv_fieldcat.
  alv_layout-colwidth_optimize = 'X'.
  alv_layout-zebra              = 'X'.

  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
      *      I_INTERFACE_CHECK                = ' '
      *      I_BUFFER_ACTIVE                  = ' '
      *      i_callback_program                = 'ZPSOL060'
      *      I_CALLBACK_PF_STATUS_SET          = ' '
      *      I_CALLBACK_USER_COMMAND           = ' '
      *      I_CALLBACK_TOP_OF_PAGE            = ' '
      *      I_CALLBACK_HTML_TOP_OF_PAGE       = ' '
      *      I_CALLBACK_HTML_END_OF_LIST       = ' '
      *      I_STRUCTURE_NAME                  =
      *      I_BACKGROUND_ID                   = ' '
      *      I_GRID_TITLE                      = 'Date Specifications'
      *      I_GRID_SETTINGS                   =
      *          is_layout                      = alv_layout
      *          it_fieldcat                    = alv_fieldcat
      *      IT_EXCLUDING                      =
      *      IT_SPECIAL_GROUPS                 =
      *      IT_SORT                           =
      *      IT_FILTER                         =
      *      IS_SEL_HIDE                       =
      *      I_DEFAULT                         = 'X'
      *      I_SAVE                            = 'A'
      *      IS_VARIANT                        = stru_disvar
      *      IT_EVENTS                        =
      *      IT_EVENT_EXIT                     =
      *      IS_PRINT                          =
      *      IS_REPREP_ID                      =
      *      I_SCREEN_START_COLUMN              = 0
      *      I_SCREEN_START_LINE                = 0
      *      I_SCREEN_END_COLUMN                = 0
      *      I_SCREEN_END_LINE                  = 0
      *      IT_ALV_GRAPHICS                    =
      *      IT_ADD_FIELDCAT                    =
      *      IT_HYPERLINK                      =
      *      IMPORTING
      *          E_EXIT_CAUSED_BY_CALLER        =
      *          ES_EXIT_CAUSED_BY_USER         =

  TABLES
      t_outtab                                = data_tab

  EXCEPTIONS
      program_error                          = 1
      OTHERS                                  = 2.

  IF sy-subrc = 1.
    WRITE: 'Program error' (002).
  ENDIF.

```

```

*-----*
*          FORM READ_HIREDATE
*-----*
*          Retrieving hiredate from infotype 0000
*-----*
FORM read_hiredate.
  CALL FUNCTION 'RP_GET_HIRE_DATE'
    EXPORTING
      persnr          = p0001-pernr
      check_infotypes = '0000'
    IMPORTING
      hiredate        = data_tab_wa-eindt
    EXCEPTIONS
      OTHERS          = 1.
ENDFORM.

```

```

*-----*
*          FORM RE548T
*          Read date specification texts
*-----*
VALUE (LANGUAGE)
* -->  VALUE (DTYPE)
*-----*
FORM re548t USING value(language)
                  value(dtype).
  CHECK t548t-sprsl NE language
    OR t548t-datar NE dtype.
  SELECT SINGLE * FROM t548t
    WHERE sprsl EQ language
    AND   datar EQ dtype.
  IF sy-subrc NE 0.
    CLEAR t548t.
  ENDIF.
ENDFORM.

```

```

*$$$- Start of INCLUDE ZPSOL050_FILL_FIELDCAT                --$$$*
*&-----*
*&   Include                ZPSOL050_FILL_FIELDCAT            *
*&-----*

*-----*
*       FORM FIELDCAT_INIT
*-----*
FORM fieldcat_init USING p_fieldcat TYPE slis_t_fieldcat_alv.
  DATA: ls_fieldcat TYPE slis_fieldcat_alv.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'PERNR'.
  ls_fieldcat-ref_tabname = 'PERNR'.
  ls_fieldcat-key       = 'X'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'ENAME'.
  ls_fieldcat-ref_tabname = 'PERNR'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'EINDT'.
  ls_fieldcat-ref_tabname = 'P0016'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'DAR'.
  ls_fieldcat-ref_fieldname = 'DAR01'.
  ls_fieldcat-ref_tabname = 'P0041'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'DTEXT'.
  ls_fieldcat-ref_tabname = 'T548T'.
  APPEND ls_fieldcat TO p_fieldcat.

  CLEAR ls_fieldcat.
  ls_fieldcat-fieldname = 'DAT'.
  ls_fieldcat-ref_fieldname = 'DAT01'.
  ls_fieldcat-ref_tabname = 'P0041'.
  APPEND ls_fieldcat TO p_fieldcat.
ENDFORM.
*$$$--- End of INCLUDE ZPSOL050_FILL_FIELDCAT                --$$$*

```





### **Contents:**

- **Exporting/importing database tables in HR**
- **Exporting/importing cluster data structures**
- **Exporting/importing with macros using a buffer**
- **Cluster authorization**
- **Data structure for payroll results**
- **Reading the cluster directory**
- **Processing the cluster directory**
- **Processing payroll results**
- **Utilities**

© SAP AG 2003



**At the conclusion of this unit, you will be able to:**

- **Import cluster data structures**
- **Process payroll results using function modules**
- **Use the logical database for payroll results**

Course Overview

**Cluster Database Tables in HR**

Data Structures in  
Personnel Administration

Data Structures in  
Personnel Planning

The Logical Databases  
PNP/PNPCE

The Logical Database PCH

Join and Projection

Customer-Specific Infotypes

Specific Commands

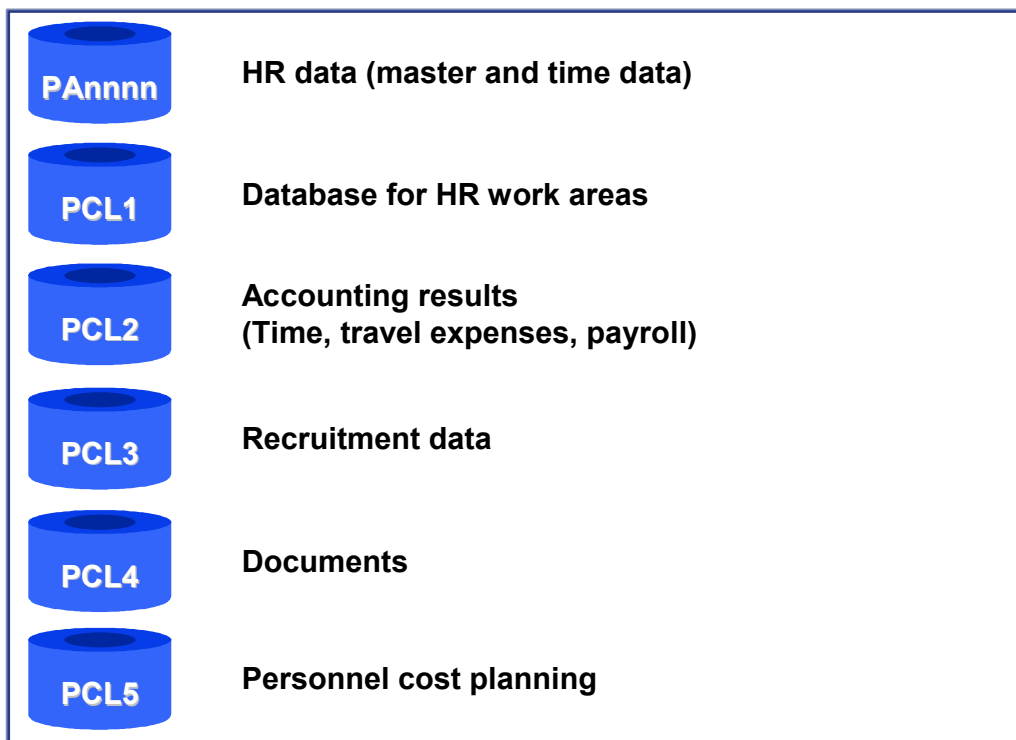
HR ABAP Features

Repetitive Structures and  
List Display

© SAP AG 2003

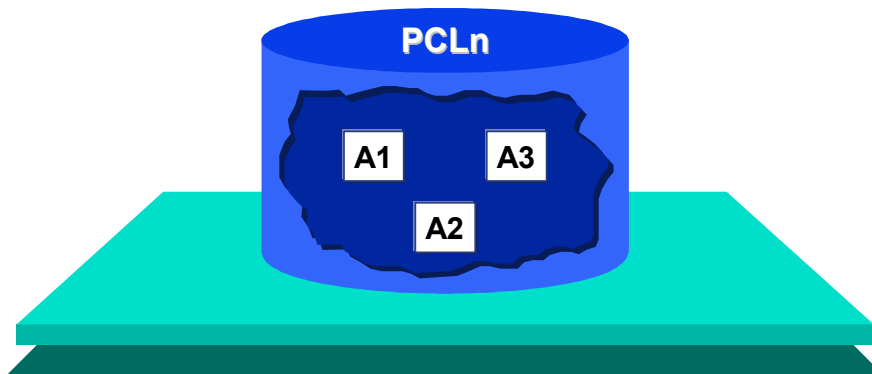


- **The different departments in your company want report lists to include the explanatory texts entered when the infotypes are maintained.**
- **The payroll department would like to use a special report to evaluate the payroll results.**



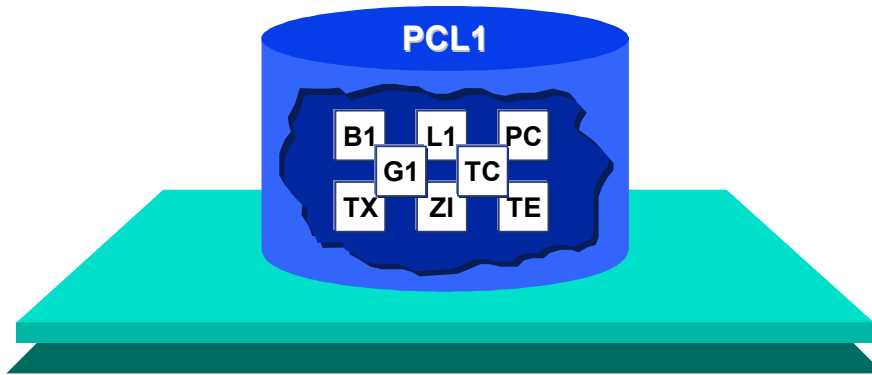
© SAP AG 2003

- The PAnnnn database tables contain all HR data sorted by infotype. They constitute the database for infotype entry screens and are evaluated by the HR logical database.
- The PCL1, PCL2, and PCL3 database tables constitute either the database for subsequent programs, such as accounting runs or evaluations, or the database for subareas within Human Resources, such as travel expenses and recruitment.
- The PCLn database tables are a type of import/export database table.



© SAP AG 2003

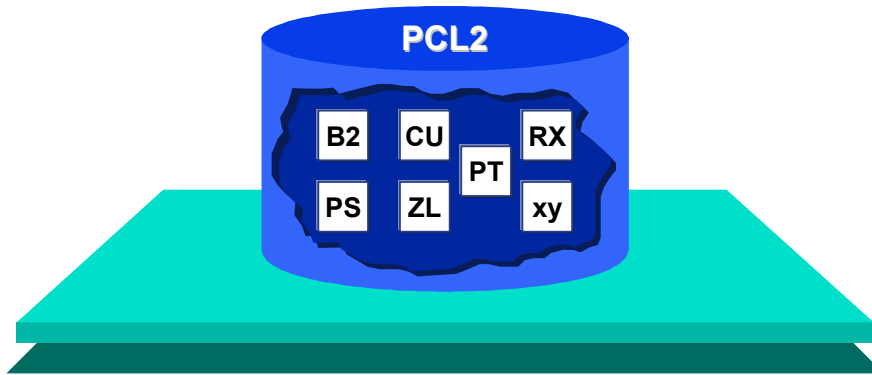
- Database tables of type PCLn are divided into subareas known as data clusters.
- Data clusters can be identified by their two-character IDs.
- The respective subareas within Human Resources work on their own cluster.
- A separate data key is defined for each subarea.
- You can use the transaction **PECLUSTER** to call a view with information on the existing clusters and the database tables in which they are stored.



© SAP AG 2003

■ The following are some of the data areas in the database table PCL1:

- B1 Time events/PDC
- G1 Group incentive wages
- L1 Individual incentive wages
- PC Personal calendar
- TE Travel expenses/Accounting results
- TC Travel expenses/Credit card data
- TX Infotype texts
- ZI Interface PDC -> Cost Accounting/Materials Management

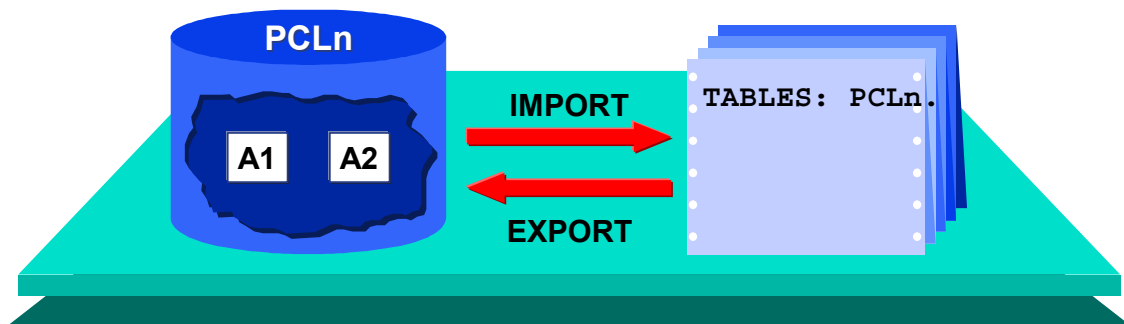


© SAP AG 2003

■ Database table PCL2 contains the following data areas:

- B2 Time accounting results
- CU Cluster directory
- CA Cluster directory for archived payroll results
- PS Generated schema
- PT Texts for generated schema
- RX Payroll results/international
- Xy Payroll results/country-specific, whereby xy represents the relation ID. This either consists of Rn (n = HR country indicator), or of the ISO code from table T500L.
- ZL Personal work schedule





© SAP AG 2003

- The PCLn cluster database tables are managed by the ABAP commands IMPORT and EXPORT.
- You can use these commands to store any data object - such as fields, structures or internal tables - on the database, or to read them from the database.
- Data is read and written using a unique key.

Field name	KEY	Length	Text
<b>CLIENT</b>	<b>X</b>	<b>3</b>	<b>Client</b>
<b>RELID</b>	<b>X</b>	<b>2</b>	<b>Relation ID</b>
<b>SRTFD</b>	<b>X</b>	<b>40</b>	<b>Work area key</b>
<b>SRTF2</b>	<b>X</b>	<b>10</b>	<b>Sort field for duplicate key</b>

#### Example of International Payroll Results:

<b>RELID</b>	<b>SRTFD</b>
RX	00001911 00001
RX	00001911 00002
RX	00001911 00003
...	

© SAP AG 2003

- The structure of PCLn database tables provides a basic structure for individual data areas.
- The name of each data area must include a two-character cluster name or relation ID.
- A key structure must also be defined; 40 bytes are available in field SRTFD for this purpose.
- In the international payroll results, for example, the field **RELID** contains the cluster name RX and the field **SRTFD** contains the eight-digit personnel number as well as a five-digit sequence number for each individual payroll result for a payroll period.
- The RELID field must contain the cluster identifier and the SRTFD field must contain the cluster key to enable the import of a record from a PCLn database table. The system fills the **SRTF2** field.
- The **CLIENT** field must only be filled if cluster data is to be imported from another client.

```
* KEY-Definition
DATA:   BEGIN OF xy-key,
        FIELD1,
        FIELDn,
        END OF xy-key.

* Definition of Data Objects
DATA:   BEGIN OF table1 OCCURS 10,
        COLUMN1,
        COLUMNn,
        END OF table1.
DATA:   BEGIN OF table2 OCCURS 10, . . .
```

© SAP AG 1999

- The data definitions of a work area are stored in separate programs using a fixed naming convention.
- They are defined as INCLUDE programs. Their names follow convention RPCnxyz0, where.
  - n = 1 or 2 for PCL1 or PCL2
  - xy = Cluster, for example, RX
  - z = 0 for international clusters or country indicator from table T500L for country-specific clusters.
  - Example: The program RPC1TX00 contains the data definition of cluster TX. In this cluster, the database table PCL1 contains the texts stored for infotypes.
- The key structure of the cluster is stored in a field string xy-KEY where the personnel number is the first element.
- The data objects (field strings and tables) are named individually per cluster.

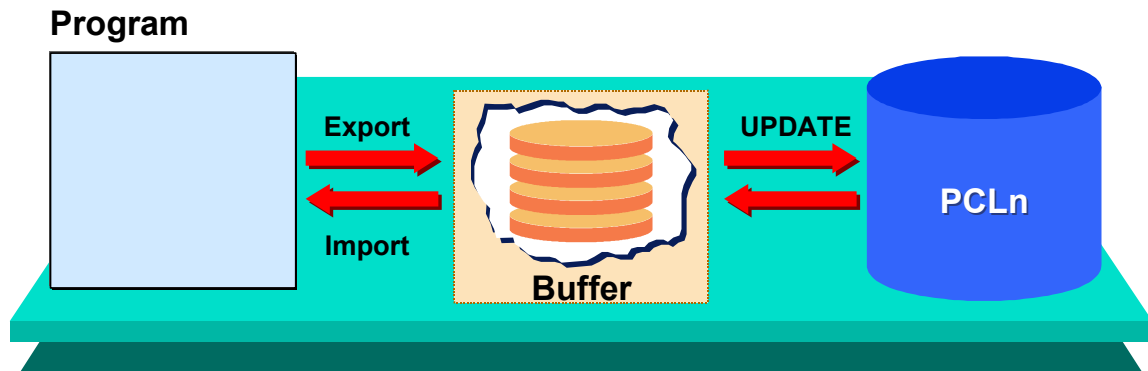
```

TABLES: PCLn,...           "Import/export table
INCLUDE: RPCnxyz0.         "Cluster definition . .

* Fill Cluster-KEY
xy-KEY-FIELD1 = <VALUE>,
. . .
* Import Record
RP-IMP-Cn-xy.
* Display data object
IF SY-SUBRC EQ 0.
    LOOP AT table1.
        WRITE. . .
    ENDLOOP.
ENDIF.
. . .
    
```

© SAP AG 2003

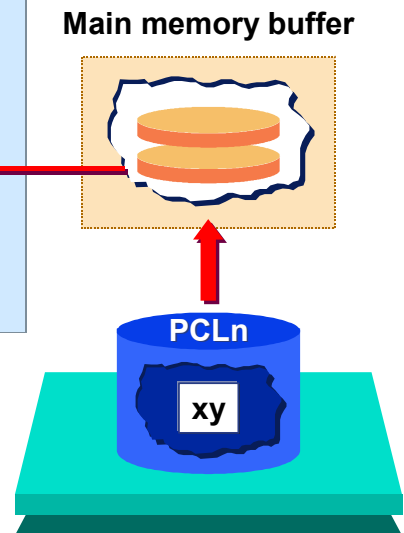
- To ensure consistency when data is exported and imported, the IMPORT/EXPORT commands are defined as macros.
- It is possible to import only a portion of the data objects in a cluster.
- The naming conventions for the macros are RP-IMP-Cn-xy and RP-EXP-Cn-xy, where n is the file name and xy the cluster name.
- If a record is read successfully, the return code is 0. If a record is not read successfully, the return code is 4.
- The macro **RP-IMP-C1-TX** for importing infotype texts to cluster TX is contained in the table Macros in ABAP Programs.
- The macros use routines that carry out two tasks:
  - Data buffering
  - Cluster authorization check



© SAP AG 2003

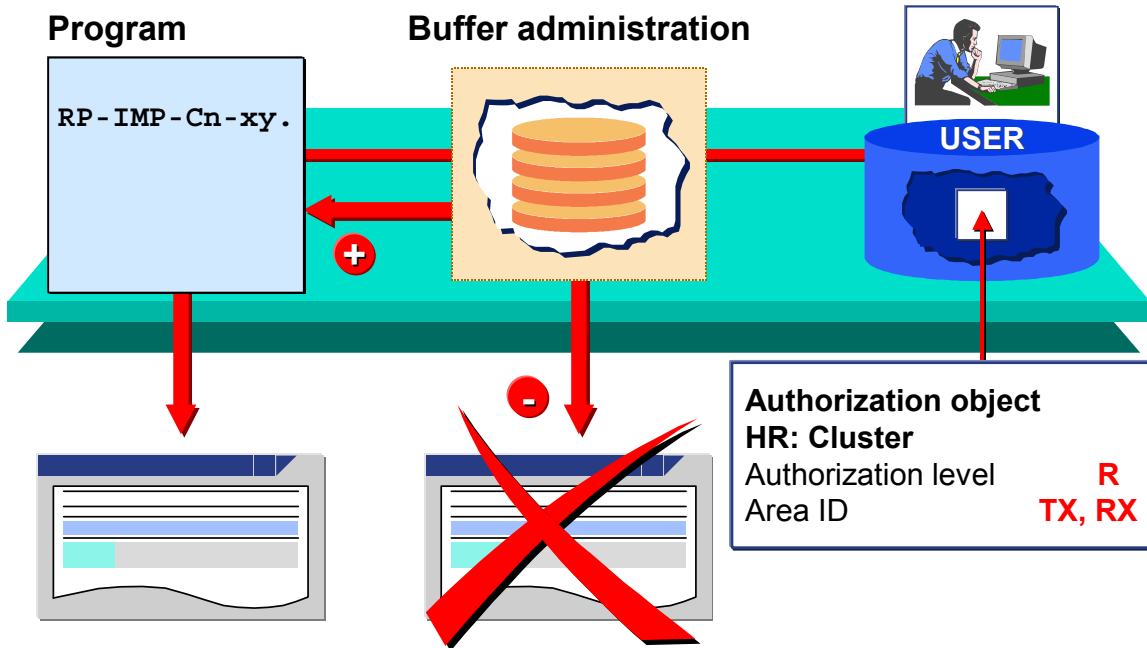
- To minimize the number of times that the database is accessed, import and export data is buffered in the main memory.
- If a test run is performed, the database is not updated. However, the payroll results of the previous period form the basis of the calculation used to determine the results of the subsequent period. For this reason, the results of a live payroll run and the results of a test run differ if test runs are performed for several periods.
- Using the buffer enables you to access the required results from the previous period.

```
* Puffer definition
INCLUDE: RPPPX00.
DATA: BEGIN OF COMMON PART buffer.
      INCLUDE RPPPX10.
DATA: END OF COMMON PART buffer.
* Fill cluster KEY
. . .
* Data import
RP-IMP-Cn-xy.
. . .
* Buffer administration routines
INCLUDE: RPPPXM00.
```



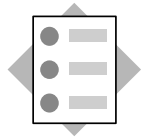
© SAP AG 2003

- If data is imported using macros, the data records are not read directly from table PCLn. Instead, the buffer directory is checked to determine whether the main memory already contains a record with the same key. If this is not the case, the record is read from PCLn to the buffer, and retrieved from the buffer by the report.
- The general data definitions for buffering are contained in the includes RPPPX00 and RPPPX10. The subroutines called by the macro must be incorporated using the include RPPPXM00.
- If data is read using a buffer, the system checks the cluster authorization. The standard import programs follow the RPCLSTxy naming convention, where xy = cluster name.



© SAP AG 2003

- If the IMPORT statement is used with a buffer, the buffer administration routines check the cluster authorizations (HR authorization object: Cluster).
- The authorization level field enables you to specify the type of access that is allowed:  
**R** provides read access, **U** provides read and write access, **S** allows data to be exported to the PCLx buffer without changing the database (simulated payroll runs).
- The *Area ID* field enables you to list the clusters for which authorization exists.



**At the conclusion of this topic, you will be able to:**

- **Use function modules to read and process payroll results**



**REPORT**  
**H99\_DISPLAY\_PAYRESULT**



## Payroll result

**Kathy Calc**      **January 19xx**

<b>WT01</b>	<b>Standard salary</b>	<b>5000 UNI</b>
<b>WT02</b>	<b>Bonus</b>	<b>300 UNI</b>
<b>WT03</b>	<b>Overtime</b>	<b>200 UNI</b>
<b>/101</b>	<b>Total gross amt</b>	<b>5500 UNI</b>

....

© SAP AG 1999

- With this report, you can display the payroll results for any country. The indicator behind each name shows whether results exist in the selected period.

Green:    Results exist

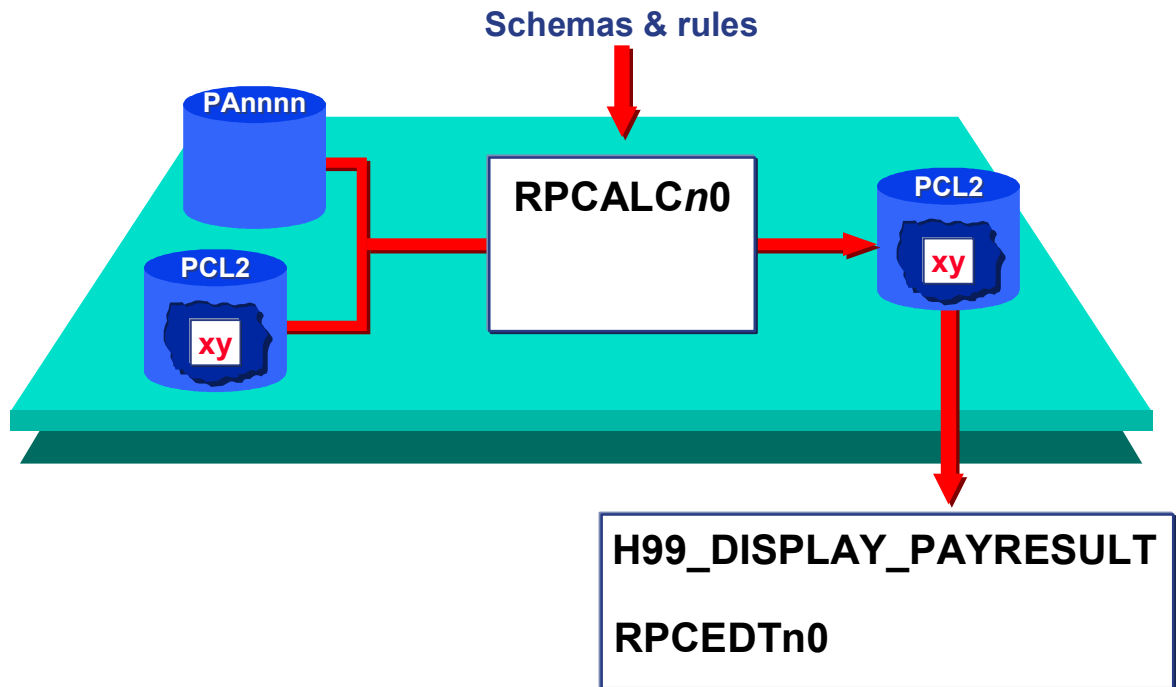
Red:      No results exist

Gray:     No authorization for displaying personnel numbers

- Payroll results are stored as structures and internal tables on the database.

- Each payroll result has a status indicator:

- A    =    Current result
- P    =    Previous result
- O    =    All other results



© SAP AG 1999

- The payroll driver, `RPCALCn0`, uses HR data (stored in the database tables `PAnnnn`) and the last payroll result (stored in the database table `PCL2`) to run the payroll for the specified period
- The program (payroll driver) imports the processing logic in the form of a schema. The schema contains functions that call the subroutines contained in the payroll driver. In many cases, the function is enhanced by rules for specific control of the subroutines.
- The payroll result generated by the payroll driver is stored in cluster `xy` of the database table `PCL2`.
- Report `H99_DISPLAY_PAYRESULT` displays the payroll results for `PCL2` and, from Release 4.6C, replaces reports `RPCLSTxy` and `HxyCLSTR`. The report is used in all country versions and the overview of payroll results is automatically displayed according to the particular country.
- Report `RPCEDTn0`, for example, lists the formatted result as a payroll form ( $n$  = HR country indicator from table `T500L`).

```

*Table containing directory of payroll results
DATA: BEGIN OF RGDIR OCCURS 100.
      INCLUDE STRUCTURE PC261.
DATA: END OF RGDIR.
DATA: COUNTRY LIKE T001P-MOLGA.
...
CALL FUNCTION 'CU_READ_RGDIR'
  EXPORTING
    PERSNR          = PERNR-PERNR
  IMPORTING
    MOLGA           = country
  TABLES
    IN_RGDIR        = RGDIR
  EXCEPTIONS
    NO_RECORD_FOUND = 1
    OTHERS          = 2.
    
```

© SAP AG 1999

- Table RGDIR contains the directory (cluster directory) for all of an employee's payroll results and is contained in cluster CU. A directory entry with the payroll area, for-period, in-period, status indicator, and the five-digit sequence number is required, together with the personnel number, to construct the key for each payroll result for an employee.
- The function module CU\_READ\_RGDIR reads table RGDIR from cluster CU. The personnel number whose payroll directory is to be read is transferred to the function module.
- If the MOLGA parameter is active, the function module returns the HR country indicator.

```

DATA: number LIKE PC261-SEQNR.
. . .
GET PERNR.
. . .
CALL FUNCTION 'CU_READ_RGDIR'
. . .

CALL FUNCTION 'CD_READ_LAST'
  EXPORTING
    BEGIN_DATE      = PN-BEGDA
    END_DATE        = PN-ENDDA
  IMPORTING
    OUT_SEQNR       = number
  TABLES
    RGDIR           = RGDIR
  EXCEPTIONS
    NO_RECORD_FOUND = 1
    OTHERS          = 2.
    
```

© SAP AG 1999

- The function module CD\_READ\_LAST determines the current payroll result for a for-period to be evaluated. To determine the correct start date and end date of the for-period, you specify the period by entering the payroll period in the selection screen. If you specify report class XXM00004 in the attributes of your report, the payroll period is entered and the start date (PN-BEGDA) and the end date (PN-ENDDA) are determined using the Payroll Periods table (T549Q).
- You enter the start and end date of the for-period for the evaluation as well as table RGDIR. The function module then gives you the sequential number (OUT\_SEQNR) for the current (A) result of the for-period.
- You can also use the following function modules:  
 CD\_READ\_PREVIOUS (reads the record that precedes the payroll record)  
 CD\_READ\_PREVIOUS\_ORIGINAL (reads the last original result that precedes the original payroll result)

```

DATA: payroll TYPE PAY99_RESULT.
DATA: rt_line TYPE LINE OF HRPAY99_RT.
. . .
CALL FUNCTION 'PYXX_READ_PAYROLL_RESULT'
  EXPORTING
    CLUSTERID                = 'RX'
    EMPLOYEENUMBER           = p0001-pernr
    SEQUENCENUMBER           = number
*   READ_ONLY_BUFFER         = ' '
*   READ_ONLY_INTERNATIONAL = ' '
*   CHECK_READ_AUTHORITY    = 'X'
    . . .
  CHANGING
    PAYROLL_RESULT           = payroll
  EXCEPTIONS
    . . .
LOOP AT payroll-INTER-RT INTO rt_line.
  WRITE: / rt_line-LGART, ...
ENDLOOP.

```

© SAP AG 2003

- You can use the function module PYXX\_READ\_PAYROLL\_RESULT to read a complete payroll result from the database table PCL2 or from the buffer. The payroll result is then transferred to the parameter PAYROLL\_RESULT. This must be declared in the calling report as a complex structure that corresponds to the structure PAY $ic$ \_RESULT ( $ic$  = ISO code).
- You can use the READ ONLY INTERNATIONAL parameter to specify that only the international part is imported. The READ\_ONLY\_BUFFER parameter prevents the database from being accessed.
- If the parameter CHECK\_READ\_AUTHORITY is activated and set to blank, the cluster authorization check is deactivated. Anonymous evaluations can then be carried out by users without cluster authorizations.

## Structure: PAY99\_RESULT

### EVP

Structure PC261

### INTER Structure PAY99 INTERNATIONAL

#### VERSC

Type PC202

#### RT

Type HRPAY99\_RT

#### CRT

Type HRPAY99\_CRT

#### BT

### NAT

Dummy  
for national  
part

```
DATA: payroll TYPE PAY99_RESULT.
DATA: rt_line TYPE LINE OF HRPAY99_RT.
. . .

* Access to accounting
result data
  WRITE: payroll-INTER-VERSC-FPPER,
  . . .
  LOOP AT payroll-INTER-RT INTO rt_line.
    WRITE: / rt_line-LGART, . . .
  ENDLOOP.
```

© SAP AG 2003

- The data structures for the international payroll results RX are described in the Dictionary in the structure PAY99\_RESULT. The structure contains the components **EVP** (directory information), **INTER** (international), and **NAT** (country-specific part). The INTER and NAT components also contain substructures such as the tables (for example, RT, and CRT) and structures (for example, VERSC) for payroll results. In cluster RX, NAT consists of a dummy field.
- The structures PAY $ic$ \_RESULT ( $ic$  = ISO code, for example, PAYUS\_RESULT for the USA) exist for the country-specific results. Here, the component NAT contains the substructures for the country-specific results.
- If you want to evaluate payroll results, you need a data structure with the type PAY $ic$ \_RESULT. For each table in the payroll results to be processed, you need a header with the type HRPAY $ic\_table\ name$  (for example, HRPAY99\_RT for the results table RT).



**At the conclusion of this topic, you will be able to:**

- **Process payroll results using the logical database for payroll results**



**Payroll results with the table WPBP  
Directory in cluster CU with table RGDIR**



**Directory information from cluster CU  
Table RGDIR**



**Data for work center and basic pay  
Table WPBP from cluster Rx**

© SAP AG 2003

- Payroll results are saved in the cluster table PCL2 to improve performance and free up memory space. However, the logical database requires some of these payroll data in a transparent form, to optimize the performance of data selection. As of the first run in Release 4.6, the payroll driver saves the data in the tables HRPY\_RGDIR and HRPY\_WPBP although this is redundant.
- However, this does not apply for payroll records created in previous releases. To make this data available for evaluation, you must start report H99U\_RGDIR\_WPBP, to fill both tables.
- The evaluation of payroll results is now supported by a logical database that is integrated in the logical database PNP for HR master data.
- The logical database allows you to select and evaluate payroll results based on a wide range of criteria. All information from a payroll record is provided in the components of a complex structure when the **GET PAYROLL** event occurs.
- To assign a report to the logical database for payroll results, enter the selection variant '900' *Selection screen for payroll results* in the report attributes. SAP delivers the sample report *EXAMPLE\_PNP\_GET\_PAYROLL*.



```
TABLES:PERNR, PYORGSCREEN, PYTIMESCREEN.
NODES:PAYROLL TYPE PAY99_RESULT.
INFOTYPES:0001.

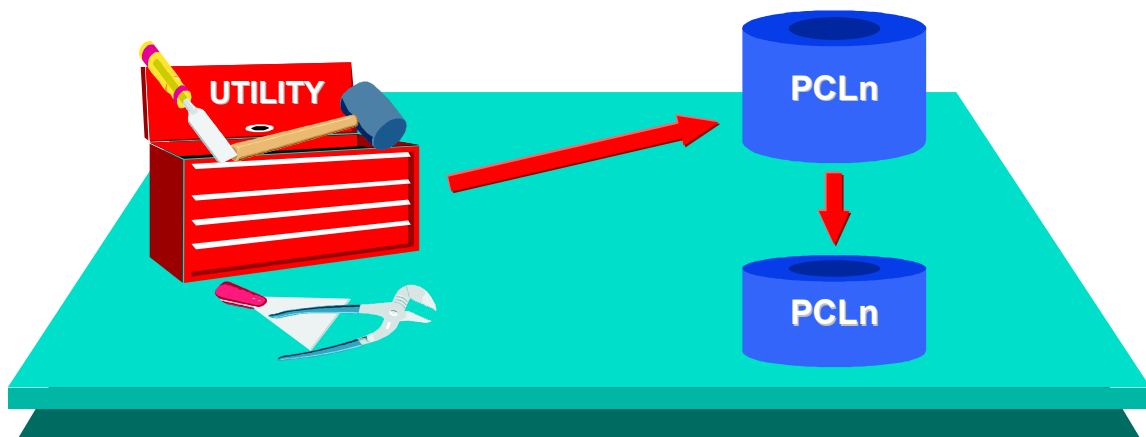
DATA: rt_line TYPE LINE OF HRPAY99_RT.
...
GET PERNR.
    RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGPS PN-ENDPS.
...

GET PAYROLL.
    LOOP AT PAYROLL-INTER-RT INTO rt_line.
        WRITE: / rt_line-LGART, . . .
    ENDLOOP.

GET PERNR LATE.
...
```

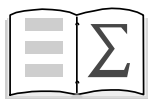
© SAP AG 2003

- PERNR, PYORGSCREEN, and PYTIMESCREEN are declared for the selection screen in the declaration for TABLES.
- NODES is used to declare the country-specific structure PAY $ic$ \_RESULT ( $ic$  = ISO code) of PAYROLL.
- The PAYROLL complex structure is filled when the **GET PAYROLL** event occurs. Note that the GET PAYROLL event only occurs for personnel numbers with a payroll result in the corresponding cluster xx. The result wage types are made available in the PAYROLL-INTER-RT table.
- The evaluation period appears in the fields PN-BEGPS and PN-ENDPS.
- The GET PERNR LATE event is triggered when all selected payroll results are processed. You can issue summaries, for example, when this event occurs.
- You can use the *Result Status* to limit the selection of payroll results.



© SAP AG 2003

- The utility transaction PU00 allows you to delete payroll results in xy clusters for specific personnel numbers.
- You can use the utility programs RPUPnD00 and RPUPnD10 to delete one or more records from any cluster. n stands for database table PCL1 or PCL2.
- You can delete payroll results using the program RPUDEL20.
- **Caution:**  
The deletion utilities should only be used for testing. Archive data to reduce the amount of data in the database.



**You are now able to:**

- **Import cluster data structures**
- **Process payroll results using function modules**
- **Use the logical database for payroll results**





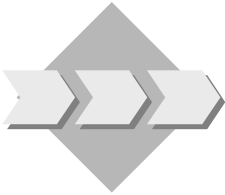
## Unit: Cluster DB Tables in HR

### Topic: Importing Cluster TX



At the end of this exercise, you will be able to:

- Import and display the texts stored in cluster TX for infotype 0040



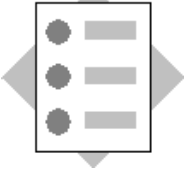
The departments at your company require a report that provides an overview of the resources loaned to employees. This information is stored in the *Objects on Loan* infotype.

- 1-1 Write a report that only lists employees who have received objects on loan. The report will also print explanations entered by the administrator in the text fields.
  - 1-1-1 The list should contain the following information for each employee:
    - Personnel number
    - Name
    - Start date
    - End date
    - Number
    - Type of object on loan
    - Description of object on loan
  - 1-1-2 Take the name of the employee from the **PERNR-ENAME** field. The object on loan corresponds to the subtype for infotype 0040. To determine the subtype text (description of object on loan), read table **T591S** with the keys infotype and subtype. Only display the first 20 characters of the employee name and subtype text.
  - 1-1-3 The include program **RPC1TX00** contains the key for cluster TX, which consists of the fields for the structure **PSKEY**. The data object for the cluster is the table **PTEXT** with the lines LINE.  
Process table P0040 with **LOOP...ENDLOOP** so that the unchanged start and end dates for the infotype records are in the header. If the field **P0040-ITXEX** is not blank, an explanatory text, which is stored in the cluster TX, is entered for the infotype record. If this is the case, enter the key and import cluster TX. If the return code for the import is not 0, display an error message.



## Unit: Exporting/Importing DB Tables in HR

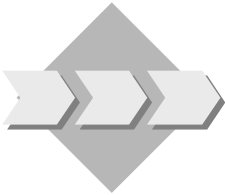
### Topic: Displaying the Last Payroll Result



At the end of this exercise, you will be able to:

- Read the cluster directory with `CU_READ_RGDIR`
- Process the cluster directory with `CD_READ_LAST`
- Read the payroll result with `PYXX_READ_PAYROLL_RESULT`
- Display the wage types from the `RT` results table

Your company's payroll department needs a list of wage types from the results table for the current (last) payroll result in a for-period.



- 1-1 Write a report to read and display the results of international payroll to cluster RX. The report should evaluate the last payroll result for the for-period. Test your program using the personnel number assigned by your trainer.

The created list should contain the following information for each employee:

Line 1:   - Personnel number  
          - Name  
          - Personnel area

Line 2:   - For-period

Line 3:   - In-period

Line 4:   - Wage type  
          - Wage type text  
          - Amount

- 1-1-1 Take the name of the employee from the **PERNR-ENAME** field.  
Information on the for-period and in-period can be found in the fields for the **VERSC** structure, which is a structure in cluster RX.

- 1-1-2 First use the function module **CU\_READ\_RGDIR** to read table `RGDIR` from cluster CU. To determine the sequential number for the last payroll result, transfer table `RGDIR` to the function module **CD\_READ\_LAST**.

- 1-1-3 Then read the payroll result for the sequential number you have determined using the function module **PYXX\_READ\_PAYROLL\_RESULT**.

- 1-1-4 The wage types and amounts are saved in the **RT** table. The texts belonging to the wage types can be found in table **T512T**.

Only display the first 15 characters of the employee name.



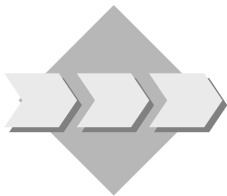
## **Unit: Exporting/Importing DB Tables in HR**

### **Topic: Logical Database for Payroll Results**



At the end of this exercise, you will be able to:

- Process payroll results using the logical database for payroll results



Your company's payroll department needs a list of wage types from the results table for payroll accounting.

- 1-1 Copy the program from the previous exercise and modify it so that you now use the logical database for payroll results.







## Unit: Cluster DB Tables in HR

### Topic: Importing Cluster TX

```
REPORT zpsol060.
```

```
TABLES: pernr,  
        pcl1,  
        pcl2,  
        t591s.                                "Subtype Table  
  
INFOTYPES: 0001,                             "Org.Assignment  
           0040.                             "Objects on Loan  
  
INCLUDE: rpc1tx00.                            "Definition Cluster TX  
INCLUDE: rpppxd00.                            "Buffer-Definition (I)  
DATA: BEGIN OF COMMON PART buffer.  
INCLUDE: rpppxd10.                            "Buffer-Definition (II)  
DATA: END   OF COMMON PART buffer.  
  
GET pernr.  
  rp_provide_from_last p0001 space pn-begda pn-endda.  
  LOOP AT p0040 WHERE endda GE pn-begda  
        AND   begda LE pn-endda.  
    PERFORM read_t591s USING p0040-infty p0040-subty.  
    PERFORM print_data.  
    IF NOT p0040-itxex IS INITIAL.  
      PERFORM read_cluster.  
      PERFORM print_cluster.  
    ENDIF.  
  ENDLOOP.  
  
END-OF-SELECTION.
```

```

*-----*
*          FORM READ_T591S
*-----*
*          Read Subtype Texts
*-----*
FORM read_t591s USING info_type sub_type.
  CHECK t591s-sprsl NE sy-langu
    OR t591s-infty NE info_type
    OR t591s-subty NE sub_type.
  SELECT SINGLE * FROM t591s
    WHERE sprsl      = sy-langu
    AND   infty      = info_type
    AND   subty      = sub_type.
ENDFORM.

```

```

*-----*
*          FORM PRINT_DATA
*-----*
FORM print_data.
  FORMAT INTENSIFIED ON.
  WRITE: / pernr-pernr,
    (20) p0001-ename,
    p0040-begda,
    p0040-enddda,
    p0040-anzkl,
    p0040-leihg,
    (20) t591s-stext.
  FORMAT INTENSIFIED OFF.
ENDFORM.

```

```

*-----*
*          FORM READ_CLUSTER
*-----*
*          Read Cluster TX
*-----*
FORM read_cluster.
  CLEAR tx-key.
  MOVE-CORRESPONDING p0040 TO tx-key.
  rp-imp-cl-tx.
  IF sy-subrc NE 0.
    WRITE: / 'Cluster TX was deleted.'(001).
  ENDIF.
ENDFORM.

```

```

*-----*
*          FORM PRINT_CLUSTER
*-----*
FORM print_cluster.
  SKIP.
  LOOP AT ptext.
    WRITE:  / ptext-line.
  ENDLOOP.
  SKIP.
ENDFORM.

*-- Subroutines
INCLUDE: rpppxm00.                                "Buffer subroutines

```



## Unit: Exporting/Importing DB Tables in HR

### Topic: Displaying the Last Payroll Result

```
REPORT zpsol070.
TABLES: pernr,
        t512t.                "Wage type texts
INFOTYPES: 0001.              "Organizational Assignment

*Table data containing directory to PCL2 payroll results file
DATA: BEGIN OF rgdir OCCURS 100.
      INCLUDE STRUCTURE pc261.
DATA: END OF rgdir.

DATA: payroll TYPE pay99_result.
DATA: rt_header TYPE LINE OF hrpay99_rt.

DATA: country LIKE t001p-molga,
      number  LIKE pc261-seqnr. "Number of last payroll result

GET pernr.
  rp_provide_from_last p0001 space pn-begda pn-endda.

  CALL FUNCTION 'CU_READ_RGDIR'
    EXPORTING
      persnr      = p0001-pernr
    IMPORTING
      molga       = country
    TABLES
      in_rgdir    = rgdir
    EXCEPTIONS
      no_record_found = 1
      OTHERS         = 2.

  IF sy-subrc = 1.
    WRITE: / 'No records found for '(001), pernr-pernr.
    REJECT.
  ENDIF.
```

```

CALL FUNCTION 'CD_READ_LAST'
  EXPORTING
    begin_date      = pn-begda
    end_date        = pn-endda
  IMPORTING
    out_seqnr       = number
  TABLES
    rgdir           = rgdir
  EXCEPTIONS
    no_record_found = 1
    OTHERS          = 2.

IF sy-subrc = 1.
  WRITE: / 'No payroll result found for'(002), pn-paper.
ELSE.
  CALL FUNCTION 'PYXX_READ_PAYROLL_RESULT'
    EXPORTING
      clusterid      = 'RX'
      employeenumber = p0001-pernr
      sequencenumber = number
  *   READ_ONLY_BUFFER      = ' '
  *   READ_ONLY_INTERNATIONAL = ' '
  *   CHECK_READ_AUTHORITY  = 'X'
  * IMPORTING
  *   VERSION_NUMBER_PAYVN   =
  *   VERSION_NUMBER_PCL2    =
  CHANGING
    payroll_result      = payroll
  EXCEPTIONS
    illegal_isocode_or_clusterid = 1
    error_generating_import      = 2
    import_mismatch_error        = 3
    subpool_dir_full             = 4
    no_read_authority            = 5
    no_record_found              = 6
    versions_do_not_match        = 7
    OTHERS                       = 8.

IF sy-subrc = 0.
  PERFORM print_rx.
ELSE.
  WRITE: / 'Result could not be read (003).
ENDIF.
ENDIF.

```

```

*-----*
*          FORM PRINT_RX                      *
*-----*
*          Print Payroll Result                *
*-----*

```

FORM print\_rx.

FORMAT INTENSIFIED ON.

WRITE: / p0001-pernr,  
     p0001-ename(15),  
     p0001-werks,  
     p0001-btrtl.

FORMAT INTENSIFIED OFF.

SKIP 1.

WRITE: / 'For period/payroll area: '(004),  
     30 payroll-inter-versc-fpper+4(2),  
     payroll-inter-versc-fpper+0(4),  
     payroll-inter-versc-abkrs,  
     / 'In-period/payroll area: '(005),  
     30 payroll-inter-versc-inper+4(2),  
     payroll-inter-versc-inper+0(4),  
     payroll-inter-versc-iabkrs.

SKIP 1.

WRITE: 'Results table: '(006).

SKIP 1.

LOOP AT payroll-inter-rt INTO rt\_line.

PERFORM re512t USING payroll-inter-versc-molga  
                     rt\_line-lgart.

WRITE: / rt\_line-lgart,  
     t512t-lgtxt,  
     rt\_line-betrq CURRENCY rt\_line-amt\_curr.

ENDLOOP.

ENDFORM.

```

*-----*
*          FORM RE512T                      *
*-----*
*          Read Wagetype Texts                *
*-----*

```

FORM re512t USING value(country\_grouping)  
                     value(wtype).

CHECK t512t-sprsl NE sy-langu  
     OR t512t-molga NE country\_grouping  
     OR t512t-lgart NE wtype.

SELECT SINGLE \* FROM t512t  
             WHERE sprsl EQ sy-langu  
             AND molga EQ country\_grouping  
             AND lgart EQ wtype.

IF sy-subrc NE 0.

CLEAR t512t.

ENDIF.

ENDFORM.



## Unit: Exporting/Importing DB Tables in HR

### Topic: Logical Database for Payroll Results

REPORT zpsol080.

TABLES: pernr, pyorgscreen, pytimescreen, t512t.

NODES: payroll TYPE pay99\_result.

INFOTYPES: 0001. "Organizational Assignment

DATA: rt\_header TYPE LINE OF hrpay99\_rt.

GET pernr.

rp\_provide\_from\_last p0001 space pn-begps pn-endps.

WRITE: / p0001-pernr,  
p0001-ename(15),  
p0001-werks,  
p0001-btrtl.

SKIP 1.

GET payroll.

WRITE: / 'For-period:'(001),  
15 payroll-inter-versc-fpper+4(2),  
payroll-inter-versc-fpper+0(4),  
/ 'In-Period: '(002),  
15 payroll-inter-versc-inper+4(2),  
payroll-inter-versc-inper+0(4).

SKIP 1.

WRITE: 'Results table: '(003).

SKIP 1.

LOOP AT payroll-inter-rt INTO rt\_line.

PERFORM re512t USING payroll-inter-versc-molga  
rt\_line-lgart.

WRITE: / rt\_line-lgart,  
t512t-lgtxt,  
rt\_line-betrg CURRENCY rt\_line-amt\_curr.

ENDLOOP.

```

*-----*
*          FORM RE512T
*-----*
*          Read Wage Type Texts
*-----*
FORM re512t USING value(country_grouping)
                value(wtype) .
CHECK t512t-sprsl NE sy-langu
      OR t512t-molga NE country_grouping
      OR t512t-lgart NE wtype.
SELECT SINGLE * FROM t512t
                WHERE sprsl EQ sy-langu
                AND   molga EQ country_grouping
                AND   lgart EQ wtype.
IF sy-subrc NE 0.
  CLEAR t512t.
ENDIF.
ENDFORM.

```



## **Contents:**

- **Data model**
- **Technical structures of infotypes in Personnel Planning**



**At the conclusion of this unit, you will be able to:**

- **Explain the data model of Personnel Planning**
- **Describe the data structures of Personnel Planning**

Course Overview

Cluster Database Tables in HR

Data Structures in Personnel Administration



**Data Structures in Personnel Planning**

The Logical Databases PNP/PNPCE

The Logical Database PCH

Join and Projection

Customer-Specific Infotypes

Specific Commands

HR ABAP Features

Repetitive Structures and List Display



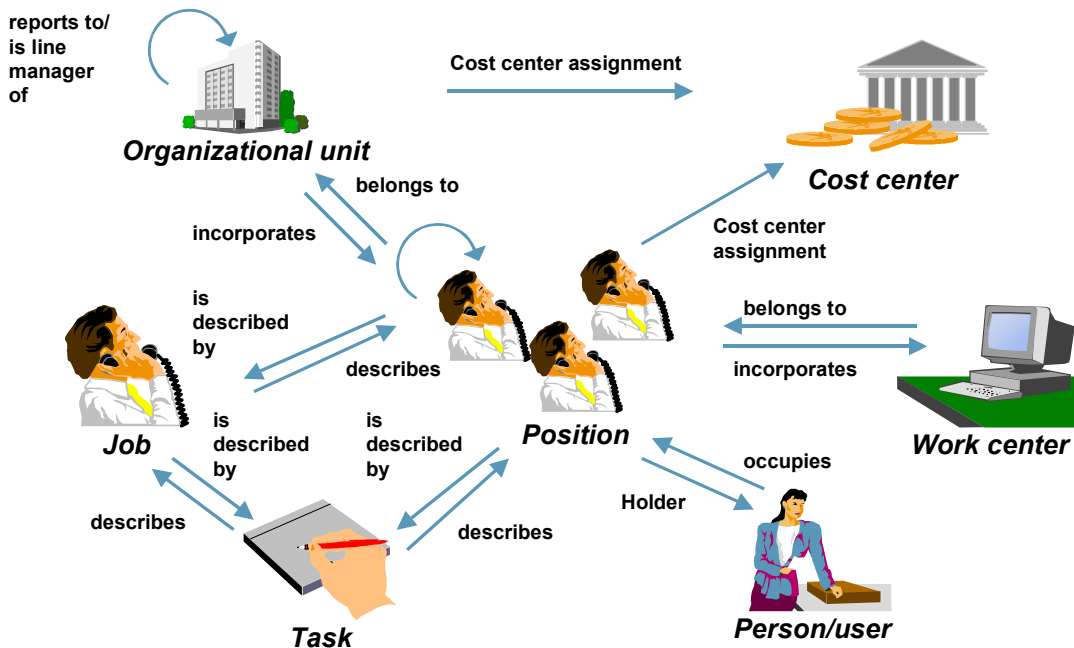
- You want to draw up your own evaluations of **Personnel Administration data**. To do this, you must learn about the data model and data structures in **Personnel Planning**.



**At the conclusion of this topic, you will be able to:**

- **Explain the Personnel Planning data model**

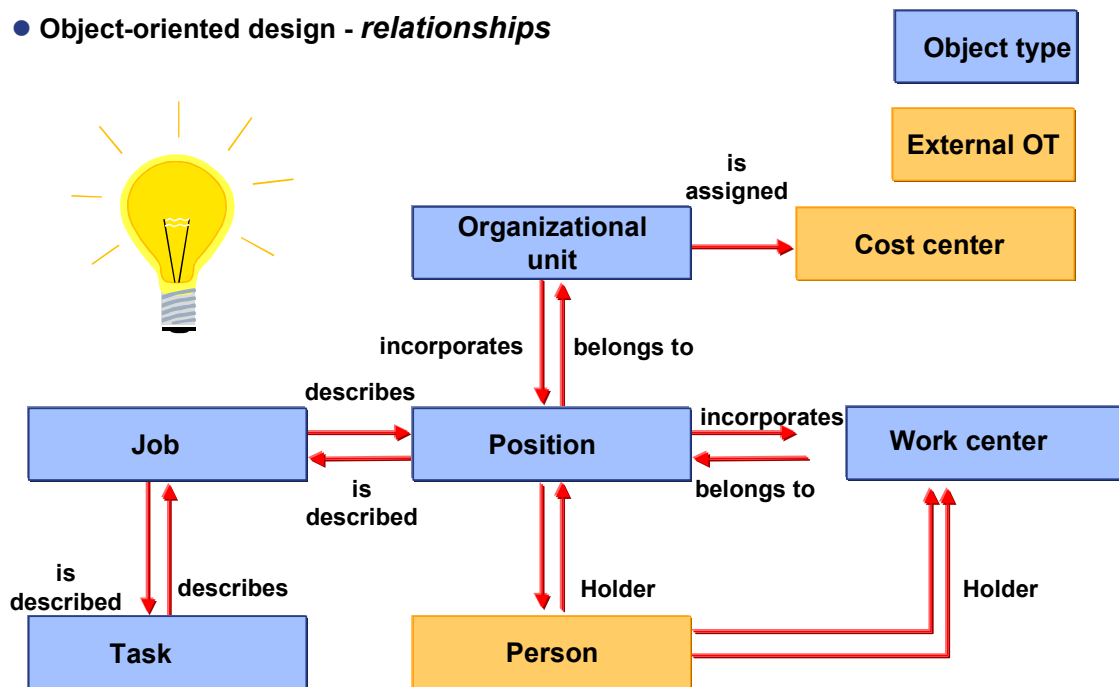
## ● Object-oriented design - *object types*



© SAP AG 2003

- The Personnel Planning data model is based on an **object-oriented design**.
- This design is based on the following concepts:
  - Object types
  - Relationships
  - Infotypes
- In Organizational Management, each element in an organization is displayed as an independent object with individual characteristics. These objects are created and maintained separately. Relationships are used to link one object to the other (see graphic). This creates a network that is flexible enough to facilitate personnel planning, projections, and evaluations.
- The cost center is an external object type because it is not maintained in Organizational Management.
- You can use Customizing to enhance the existing data model by defining new object types, for example, and establishing new relationships between the various object types. Each standard object type consists of two letters, and the customer namespace is 00 to 99.
- This data model (object types and relationships) also constitutes the basis of other applications within Personnel Planning, such as Training and Event Management (business event hierarchies) and Personnel Development (for example, qualification catalog).

- Object-oriented design - *relationships*



© SAP AG 2003

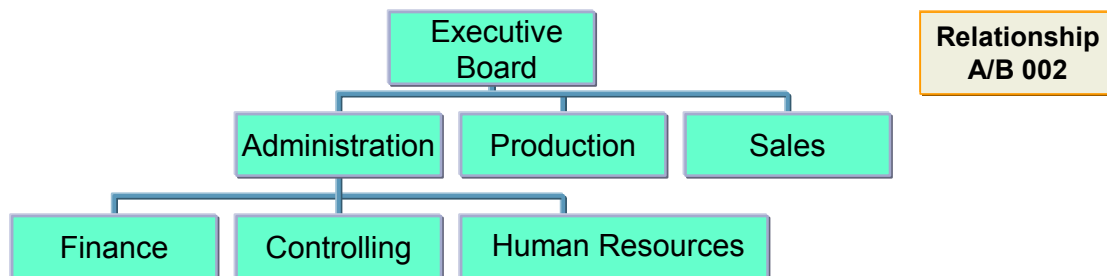
- You can use relationships to define links between individual object types in the system.
- Relationships are usually defined in two directions: *A = bottom up* and *B = top down*.
- For example: Relationship "003" ("incorporates/belongs to") is defined in the following directions:
  - *A003 "belongs to"* and *B003 "incorporates"*
- However, this convention is an optional suggestion that you may choose to adopt when you first create a relationship. It is crucial that you do not change or invert a relationship to directions A and B at a later stage.
- For internal object types, each relationship is also stored physically on the database in both directions. For relationships to external object types, only one direction may be stored on the database, depending on the key structure of the external object type. In this case, only the relationship to the external object is stored (**external object types without the inverse relationship**).

**An organizational unit "reports to" another organizational unit**

For example: Finance "reports to" Administration

**An organizational unit "is the line supervisor of" other organizational units**

For example: Executive Board "is the line supervisor of" Administration, Production, and Sales

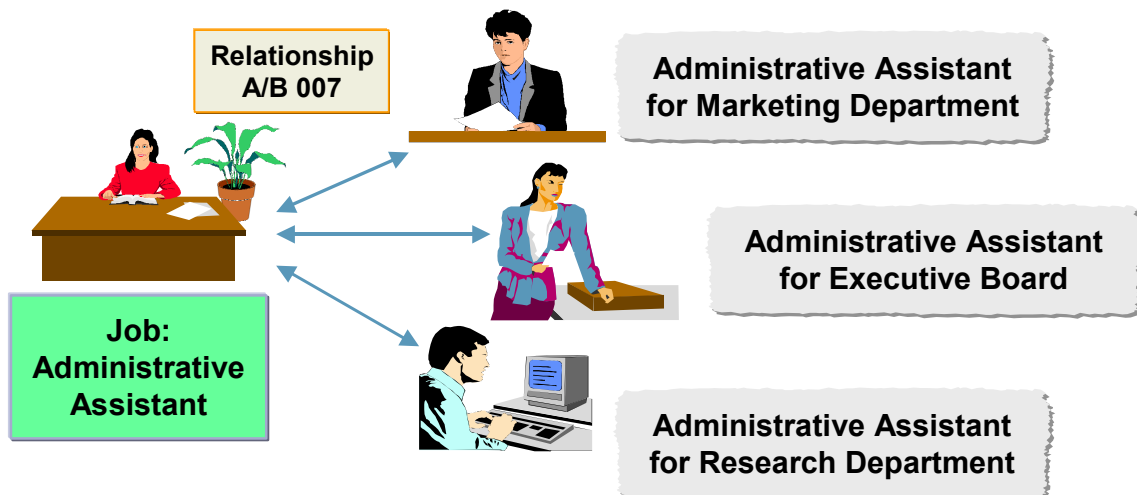


© SAP AG 2003

- Relationships are usually stored on the database in two directions.



**The job "describes" the position.  
The position "is described by" the job.**

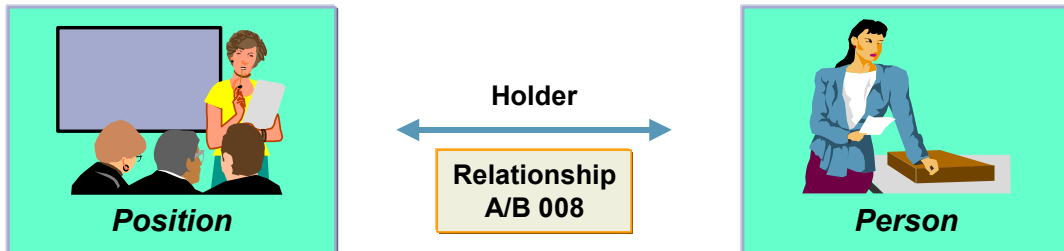


© SAP AG 2003

- The term "job" is used as a general classification, such as "secretary" or "head of department".
- More than one employee can have the same job. For example, twenty employees can have the job of secretary.
- A position refers to the individual assignment of an employee in the company, for example, sales manager, secretary of the marketing departments. You create positions and establish relationships between them to map the line structure (organization chart) in your company.

### The position is assigned to a person

For example: Ms Smith is the "holder" of the position of HR manager.



**The relationship between position and person creates integration in master data.**

© SAP AG 2003

- If Organizational Management is actively integrated with master data, the holder relationship between position and person is also stored for the person in infotype 0001 *Organizational Assignment*, although this is redundant.

**Object characteristics are maintained in the form of infotypes.**

## Main characteristics

- **Object** (1000)
- **Relationship** (1001)

## Other characteristics

- **Description** (1002)
- **Working time** (1011)
- **Cost planning** (1015)
- **Address** (1028)

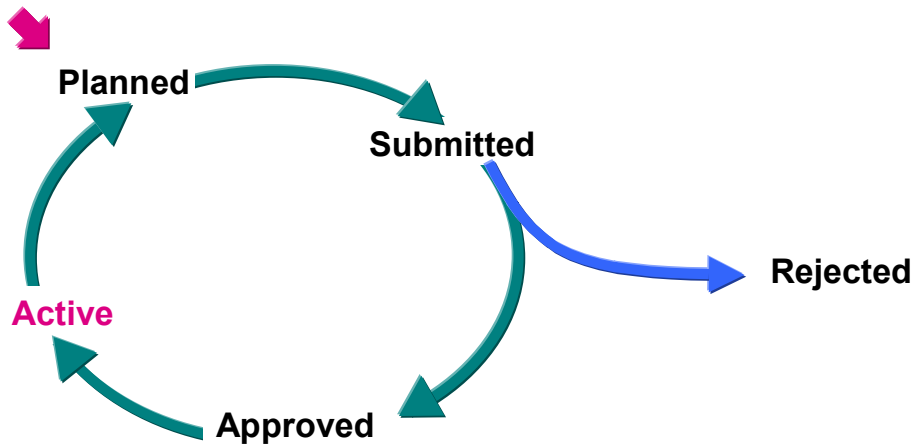
## Examples

- **Organizational unit, position, job**
- **Belongs to, reports to**
- **General text**
- **Planned working hours**
- **Cost elements**
- **Main address, second address**

© SAP AG 2003

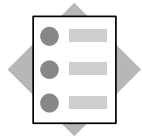
- The *Object* infotype performs key functions. It enables you to create new objects, such as organizational units and jobs, and determines the validity period of all other infotypes that describe the characteristics of existing objects.
- The *Object* infotype (1000) defines an object's existence. You can use it to store a short text and a long text for the object.  
An object is uniquely identified by the following:
  - Plan version
  - Object type
  - Object identification (ID number)
- Once you have used this infotype to create an object, you can use other infotypes such as the *Relationships* (1001) and *Description* (1002) to describe its characteristics.
- Records stored in the *Relationships* infotype (1001) serve as a descriptive and functional link between objects. For example:
  - A person occupies a position.
  - A position belongs to an organizational unit.
  - An organizational unit is assigned to a cost center in CO.

- **Object** (Infotype 1000)
- **Relationship** (Infotype 1001)



© SAP AG 2003

- One of the following statuses is assigned to each object/infotype:
- **Active** means that an object or infotype record can be currently be used.
- **Planned** means that an object or infotype record has been proposed but is not yet active, which means it cannot be used.
- **Submitted** means that an object or infotype record is currently being checked by a person or group of persons before it is approved or rejected.
- **Approved** means that a submitted object/infotype record has been accepted.
- **Rejected** means that a submitted object/infotype record has not been approved.
- You can create objects with the status *Planned* or *Active*.

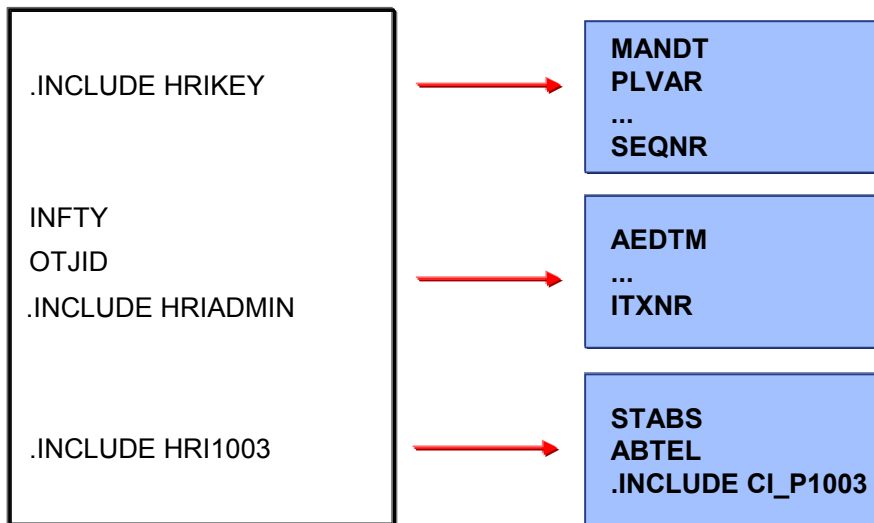


**At the conclusion of this topic, you will be able to:**

- **Describe the data structures of Personnel Planning**

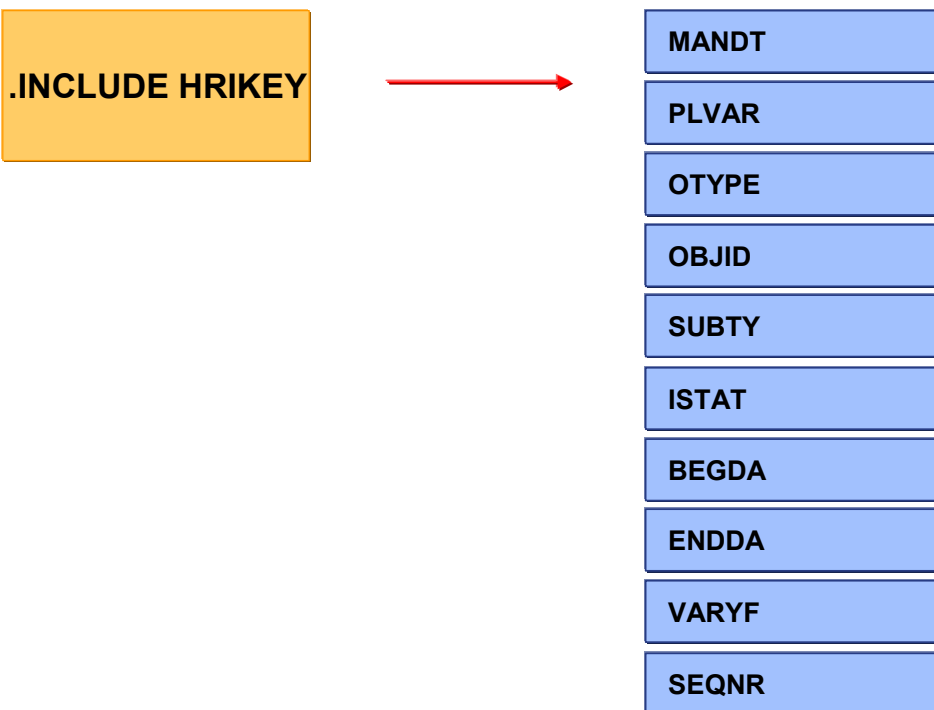
- Transparent tables *Dictionary field definition*

Example: HRP1003



© SAP AG 2003

- With the exception of HRP1000 and HRP1001, each infotype table HRPnnnn is defined in the Dictionary primarily using the following three include structures:
  - **PAKEY:**  
Contains the infotype table key fields
  - **HRIADMIN:**  
Contains administrative information on the infotype record (amongst other things, the date the last change was made and the name of the user who made the change)
  - **HRInnnn:**  
Contains the infotype-specific fields that holds the infotype's actual information and data
- The structure HRInnnn is also included by the structure Pnnnn so that the transparent table HRPnnnn and the logical structure Pnnnn are always consistent with each other.



© SAP AG 2003

- The primary key of the transparent table `HRPnnnn` is made up of the following key fields:
  - **MANDT**: A participant in the system that is legally and organizationally self-contained
  - **PLVAR**: One- or two-digit alphanumeric **plan version key** that distinguishes different organizational plans
  - **OTYPE**: One- or two-digit alphanumeric **object type key** (for example, 'S' for position, 'Q' for qualification, 'E' for event)
  - **OBJID**: Eight-digit numerical, non-mnemonic key that displays an individual object (for example, an organizational unit, a qualification or an event)
  - **SUBTY**: **Subtypes** are subdivisions of infotypes. The subtypes of an infotype can have different time constraints and create their own history.
  - **ISTAT**: One-digit numeric key that represents a planning status. All objects and their descriptive infotypes have a status. You can use this status to guide objects and infotypes through a planning cycle.
  - **BEGDA**: Valid from date
  - **ENDDA**: Valid to date
  - **VARYF**: Variation field (contains the target object for relationships)
  - **SEQNR**: The **sequential number** allows you to differentiate infotype records that have the same key and the time constraint '3' (any number of valid data records of an infotype at any one time). It is automatically assigned by the system.

**.INCLUDE HRIKEYL**



MANDT

PLVAR

OTYPE

OBJID

SUBTY

ISTAT

BEGDA

ENDDA

LANGU

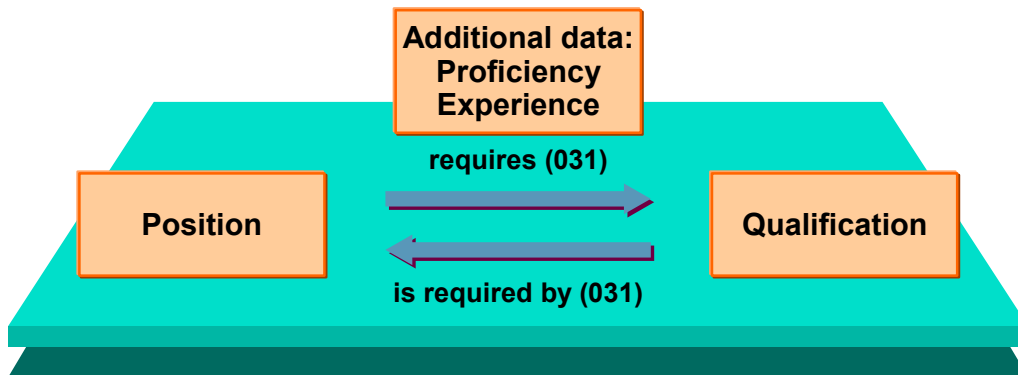
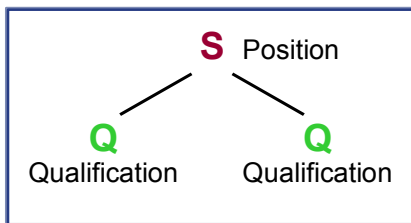
DUMMY

SEQNR

© SAP AG 2003

- **Language-specific infotypes** are infotypes that contain one or more text fields (TEXT type) that can be translated.
- A language-specific infotype is labeled as such in the *LANGU* ("Language-specific") field in table T777D.
- Examples of language-specific infotypes in the standard system include:
  - Infotype 1000 (Object): Object short and long texts can be translated
  - Infotype 1002 (Description): A text of any length can be translated
- To store the language of the infotype record in language-specific infotype records, an easily changeable primary key structure (*HRIKEYL*) is used for language-specific infotypes, compared with the non-language-specific infotype (*HRIKEY*). The language of the infotype record is stored in the first position of the VARYF field, that is, instead of the ten-digit VARYF field, the *HRIKEYL* primary key structure contains the following fields:
  - *LANGU* (language ID) with length of 1
  - *FILLER* (filler field) with a length of 9
    - In this way, the key of a language-specific infotype record differs from the key of its translated infotype records in the *LANGU* field. All other key fields are identical.



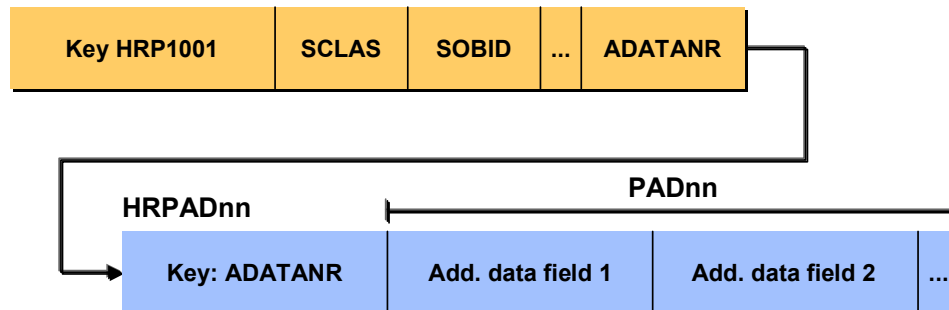


© SAP AG 2003

- Additional data on relationships comprises attributes that cannot be stored in isolation as infotypes for the objects in question because they refer exclusively to the concrete relationship between two objects.
- For example:  
The relationship 031 "requires/is required by" between a position and a qualification. A position may require the *Language skills* qualification, with the qualification providing no information on the level of language skills. The proficiency of language skills is stored as additional information in the relationship between the position and the qualification.

- **Technical implementation - *additional data***

Infotype **HRP1001**



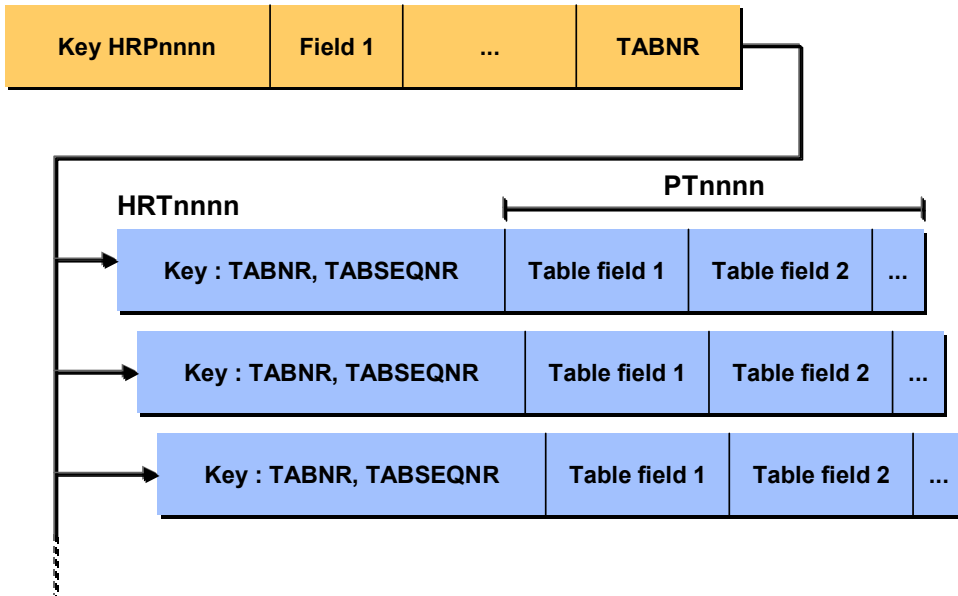
**Table T77AD:** Name of database table for an additional data structure

© SAP AG 2003

- As the structure of additional data depends on the type of relationship, the additional data of the *PADnn* structure must be stored in the separate table *HRPADnn*.
- Table T77AD is used to store the assignment between the database table and the additional data structure.
- The relationship between the relationship record (table *HRP1001*) and the additional data record (table *HRPADnn*) is determined by the **additional data pointer** (field *ADATANR*). The additional data pointer is determined internally by SAP using internal number distribution and cannot be changed externally.
- In the same way as for infotype records, the logical structure *PADnn* should always be used when you work with additional data, to avoid direct database accesses of table *HRPADnn*.
- In the *P1001* logical structure of a *relationship record* (infotype 1001), the additional data is stored in the *ADATA* field. The content of the *P1001-ADATA* field is transferred depending on the type of relationship in the corresponding additional data structure *PADnn* (in accordance with table *T77AR*).

- **Technical implementation - *table infotypes***

Table infotype **HRPnnnn**



© SAP AG 2003

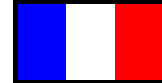
- **Table infotypes** refer to infotypes whose data has a repetitive structure of any length (for example, text with any number of lines, business event schedules with any number of days). To save repetitions of any length, the data part must be stored with the repetitive structure ("table part", described by the *PTnnnn* logical structure) in the separate table *HRTnnnn*. The name of this transparent table is stored in the table *T777D* in the *TBTAB* field ("Database table for table infotype"). In this way, the *T777D-TBTAB* field identifies a table infotype.
- The relationship between the primary record (table *HRPnnnn*) and the table part (table *HRTnnnn*) is determined by a **table pointer** (*TABNR* field). In the same way as the additional data pointer, the table pointer is determined internally by SAP using internal number distribution and cannot be changed externally.
- In the same way as for infotype records, when you are working with table infotypes, the *PTnnnn* logical structure should always be used for the *table part*, to avoid direct database accesses of *table HRTnnnn*.

- **Definition *country-specific infotypes***

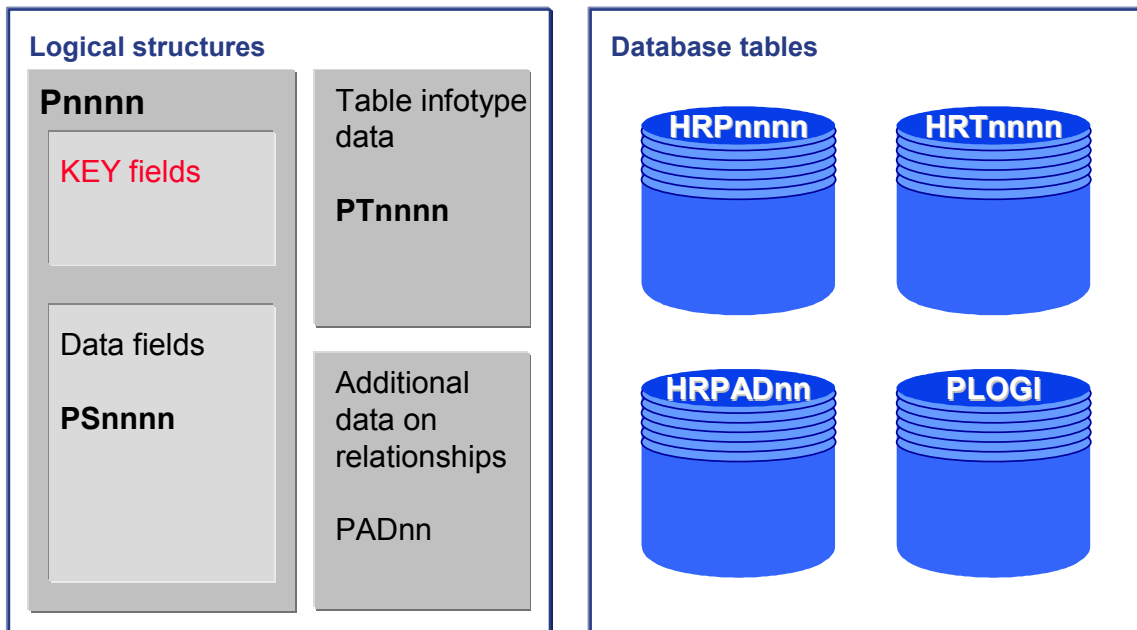
- ➔ **Country-specific infotypes** = Infotypes that are only relevant for a specific country and should therefore only appear when the corresponding country indicator is entered in general infotype overviews

- ➔ **Table T77NI** (IMG path: "Infotype maintenance" - "Maintain country-specific infotypes")

Infotype	Country ID
1600	FR
1601	FR
1610	US
1611	US
...	

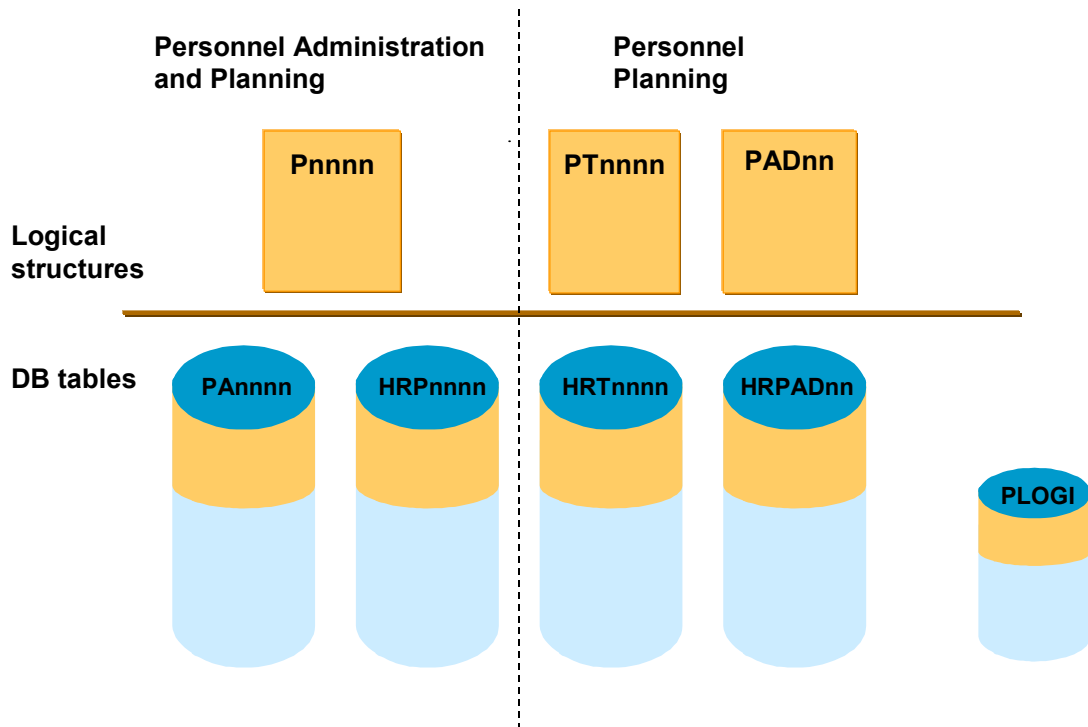


- The *NAT\_INFITY* ("Country-specific infotype") field is used to label a country-specific infotype in table *T777D*.
- The country-specific infotype is assigned to one or more country IDs (from table *T005*) in the table *T77NI*.
- When you enter the country ID in detailed maintenance (using the corresponding dialog box) and in the report *RHDESC00* (using a parameter), the corresponding country-specific infotypes are also displayed in infotype overviews.
- You can use the *PNI SPA/GPA* parameter to set user-specific default settings for the country ID. You can store up to six country IDs here. You can use the *PNALL SPA/GPA* parameter to activate the general display of all country-specific infotypes.



© SAP AG 2003

- The **Pnnnn** logical structure contains an infotype's key and data fields. During interface definition, it is used in Personnel Administration between programs and between programs and screens.
- The **PTnnnn** structure contains the table infotype data fields for table infotypes.
- The **PADnn** structure defines additional data for specific relationships.
- The transparent **HRPnnnn** tables contain the infotypes used in Personnel Planning. The table parts of table infotypes are stored in the **HRTnnnn** tables. Additional data on relationships is stored in the **HRPADnn** tables.
- The transparent table **PLOGI** contains an index of all objects on the Personnel Planning database. You can use the **RHPLOGI0** report for display/set up. This object directory forms the basis of all evaluations using the **PCH** logical database.



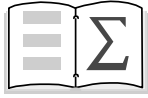
© SAP AG 2003

- The following elements exist for each nnnn infotype:
  - A *Pnnnn* logical structure
  - A *PAnnnn* transparent table (administration infotype) or *HRPnnnn* (planning infotype)
- The following elements also exist for each table infotype nnnn in Personnel Planning:- A logical structure *PTnnnn* for the table part- A *HRTnnnn* transparent table for the table part
- Each relationship type with additional data is also assigned the following:
  - A *PADnn* logical structure for additional data of this relationship type
  - A *HRPADnn* transparent table for the additional data of this relationship type
- The logical structures are crucial for programming in HR. Direct accesses to transparent tables are not required and are therefore to be avoided.
- The *PLOGI* transparent table provides a redundant directory of all Personnel Planning objects for the 1000 infotype and is used for evaluations.
- For each Personnel Planning object, the table contains just one entry with the following object ID:
- MANDT (Client)/PLVAR (Plan version)/OTYPE (Object type)/OBJID (Object ID)
- You can use the *RHPLOGI0* report to display and set up the table.

Element	Definition	Included	Changeability
<b>HRIKEY(L)</b>	Structure	-	No
<b>HRInnnn</b>	Structure	CI_Pnnnn	HR10nnn-HR18nnn: Only CI_Pnnnn HR19nnn: Yes
<b>Pnnnn</b>	Structure	HRInnnn	P0nnn-P8nnn: No Only HR19nnn
<b>HRPnnnn</b>	Transparent table	HRIKEY or HRIKEYL	HRP0nnn-HRP8nnn: No Only HR19nnn
<b>PTnnnn</b>	Structure	-	PT0nnn-PT8nnn: No PT9nnn: Yes
<b>HRTnnnn</b>	Transparent table	PTnnnn	HRT0nnn-HRT8nnn: No Only PT9nnn
<b>CI_Pnnnn</b>	Structure	-	Yes

© SAP AG 2003

- The table displayed above provides an overview of the data structures and transparent tables that belong to an infotype in Personnel Administration.



**You are now able to:**

- **Explain the Personnel Planning data model**
- **Describe the data structures of Personnel Planning**



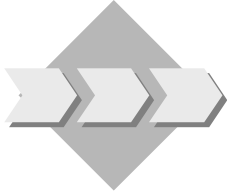


## Unit: Data Structures in Personnel Planning



At the end of this exercise, you will be able to:

- Explain the Personnel Planning data model
- Describe the data structures of Personnel Planning



You want to draw up your own evaluations of Personnel Administration data. To do this, you must learn about the data model and data structures in Personnel Planning.

1-1 The Personnel Planning data model is based on an **object-oriented design**.

1-1-1 Name the concepts that form the basis of this design:

---

---

1-1-2 How can you extend the Personnel Planning data model?

---

---

1-2 What is meant by additional data on relationships?

---

---

1-2-1 In which logical structure is additional data stored?

---

---

1-3 What is meant by a table infotype?

---

---

1-3-1 Which fields does the PTnnnn structure contain?

---

---





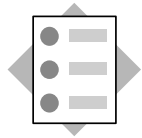
## Unit: Data Structures in Personnel Planning

- 1-1 The Personnel Planning data model is based on an **object-oriented design**.
  - 1-1-1 The following concepts form the basis of this design:
    - Object types
    - Relationships
    - Infotypes
  - 1-1-2 You can extend the Personnel Planning data model by defining new object types and new relationships in Customizing.
- 1-2 Additional data on relationships is made up of attributes that cannot be stored in isolation for the objects in question because they refer exclusively to the concrete relationship between two objects.
  - 1-2-1 The additional data is stored in the P1001 logical structure in the ADATA field.
- 1-3 A table infotype is an infotype whose data part has a repetitive structure.
  - 1-3-1 The PTnnnn structure contains the repetitive data fields for table infotypes.



## **Contents:**

- **Sequential and structural evaluations**
- **Evaluation paths**
- **Structural parameters and structural conditions**
- **Available internal information**



**At the conclusion of this unit, you will be able to:**

- **Explain the functions of the logical database PCH**
- **Create reports for sequential evaluations of the organizational structure**
- **Create reports for structural evaluations of the organizational structure**
- **Evaluate additional data on relationships**
- **Evaluate table infotypes**

Course Overview

Data Structures in  
Personnel Administration

The Logical Databases  
PNP/PNPCE

Join and Projection

Specific Commands

Repetitive Structures and  
List Display

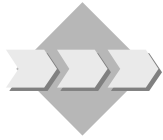
Cluster Database Tables in  
HR

Data Structures in  
Personnel Planning

 **The Logical Database PCH**

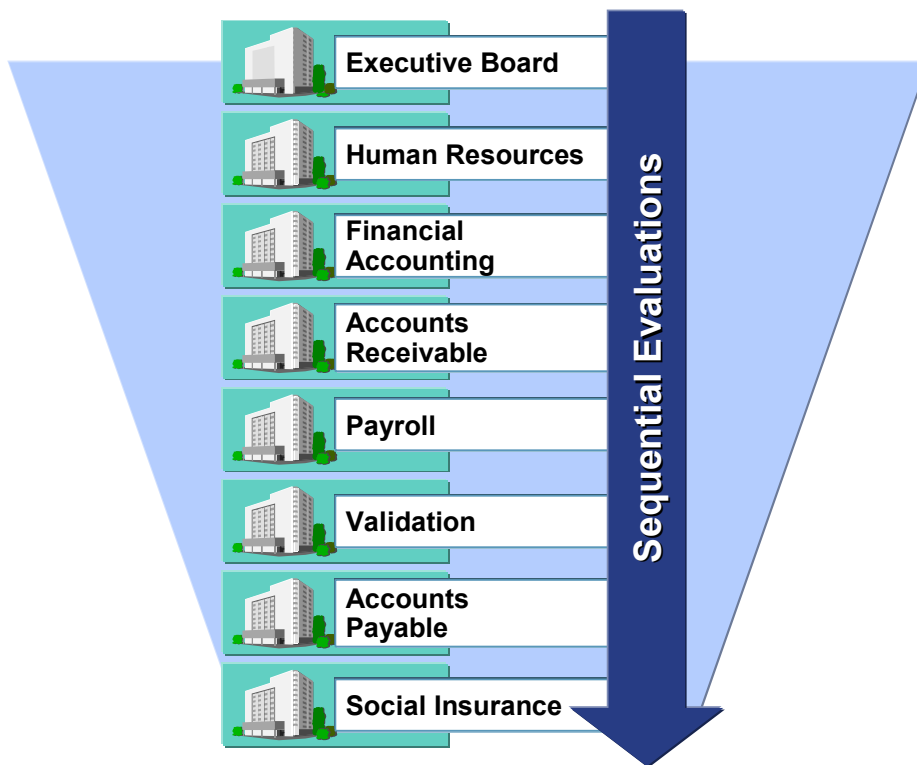
Customer-Specific Infotypes

HR ABAP Features



- **Your company's Executive Board requires an overview of the staff positions with the names of the employees holding these positions, based on the organizational structure.**

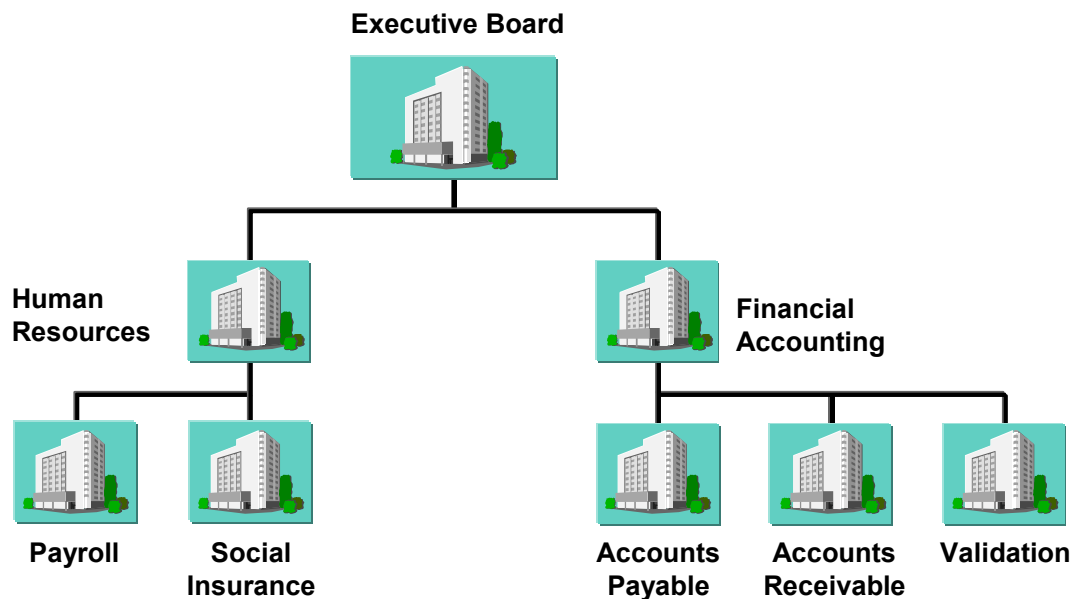




© SAP AG 2003

- You can specify objects for a sequential evaluation using their IDs. The sequential evaluation then takes place for all of the objects you have specified. For example, you can display a list of all organizational units at your company.

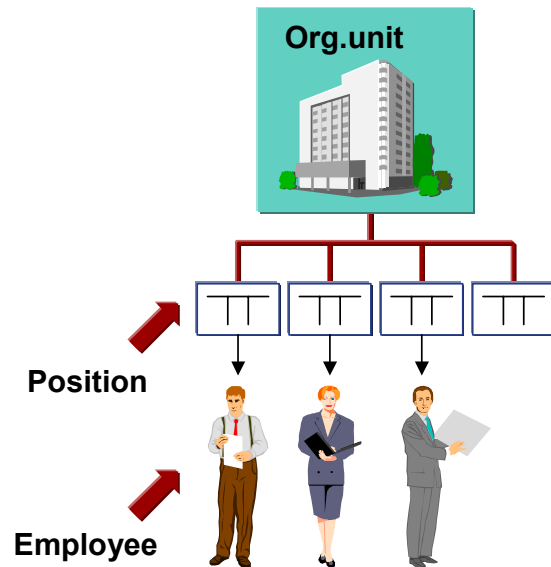
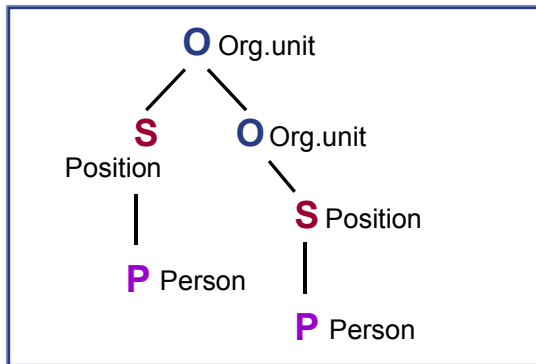
- When structural evaluations are performed, reports must take evaluation paths into account



© SAP AG 2003

- When a structural evaluation is performed, the objects to be evaluated are also listed. However, the system interprets these entries differently. It treats each selected object as a root object and uses it as a starting point for a hierarchical structure, which it builds up using a specific evaluation path.
- The evaluation path consists of a series of relationships to be evaluated, starting from the root object.
- Structural reports can be displayed using structural graphics.

- Organizational units O
- Jobs C
- Positions S
- Persons P



© SAP AG 2003

- An evaluation path describes a set of relationships between objects in a hierarchical structure. Evaluation path **O-S-P**, for example, describes the set of relationships found between organizational units, positions, and persons.
- Evaluation paths are used to select objects for structural evaluations. You choose an evaluation path, and the system evaluates the structure along the evaluation path. The report only evaluates objects that it finds in the specified evaluation path.
- Every standard report usually has a defined standard evaluation path. They are set in the system and must not be changed. The standard selection screen enables you to choose evaluation paths. You can also create new evaluation paths to meet your company's particular requirements.
- Report **RHWEGID0** displays all of the evaluation paths defined in the system, between the starting object type, and the target object type.

## Evaluation path O-S-P

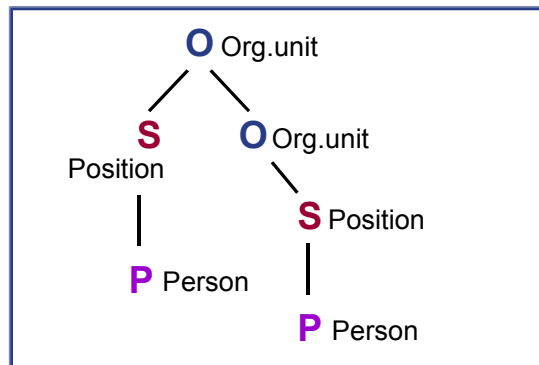
## Occupation of positions in line with organizational structure



No.	OT	A/B	Relationship	Priority	Type	Linked object	Skip
10	O	B	003 incorporates	*	S		
20	S	A	008 holder	*	P		
30	O	B	002 is line supervisor of	*	O		

### A/B

A = bottom up  
B = top down



© SAP AG 2003

- This evaluation path determines all of the assigned positions (S) and their holders (P) for a specific organizational unit (O). The subordinate organizational units are processed in exactly the same way.
- The "Skip" field enables you to determine that a specific relationship within an evaluation path is included in the evaluation but not displayed in the report list.
- Some evaluation paths consist of just one relationship. For example, **A001** is a subdivision of and **B001** is subdivided into. Thus, each relationship in the standard system has two evaluation paths.
- The convention A = bottom up and B = top down can be taken into account when a relationship is first defined. However, this is not an obligatory rule, and cannot be changed at a later stage.

```

TABLES:      OBJEC.
INFOTYPES:  0002,      "Personal data
            1003,      "Department/function
            ....
            nnnn.

GET OBJEC.
  LOOP AT Pnnnn
    WHERE BEGDA LE PC-ENDDDA
    AND   ENDDA GE PC-BEGDA.
    WRITE...
  ENDLOOP.
    
```

© SAP AG 2003

- To link a report with the **PCH** logical database, this database must be specified in the attributes.
- Use the TABLES statement to determine whether a sequential or structural evaluation is carried out. A sequential evaluation is carried out in the example outlined above.
- For each selected object, internal table Pnnnn is filled per infotype with all available infotype records. The internal tables can then be processed in a LOOP or PROVIDE.
- The infotypes from Personnel Administration can also be imported.
- In line with the PN structure of the PNP logical database, the PC structure is filled from the selection screen. PC-BEGDA and PC-ENDDDA contain the data selection period.
- When the GET event occurs, all information from the start of the object period for the infotype 1000 is in the **OBJEC** structure.

```
TABLES: OBJEC, GDSTR.  
INFOTYPES: 0002,      "Personal data  
            1003,      "Department/function  
            ....  
            nnnn.  
  
GET OBJEC.  
  LOOP AT Pnnnn  
    WHERE BEGDA LE PC-ENDDA  
    AND   ENDDA GE PC-BEGDA.  
    WRITE...  
  ENDLOOP.
```

© SAP AG 2003

- The only difference between a structural and sequential evaluation is the additional **GDSTR** entry in the TABLES statement. This ensures that the structure parameters are shown on the standard selection screen.
- **Note:**  
If you do not specify an evaluation path before starting the program, a sequential evaluation is performed.

### INITIALIZATION.

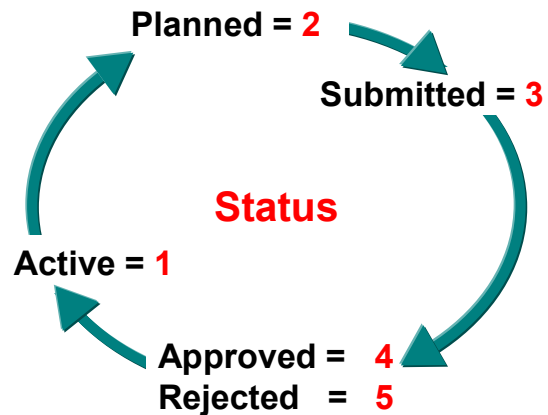
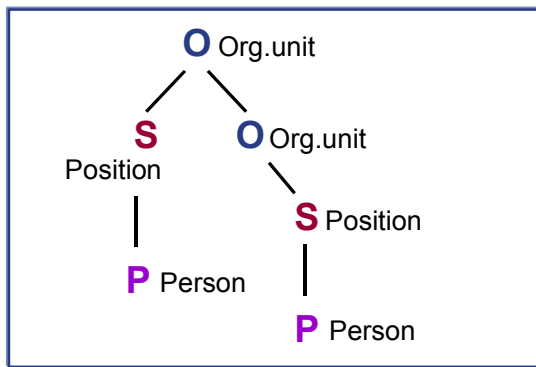
```
PCHPLVAR      = '01'.
PCHOTYPE      = 'O'.
PCHENDDA      = SY-DATUM.
PCHWEGID      = 'O-S-P'.
PCHOBJID-LOW  = '00000001'.
APPEND PCHOBJID.
```

### GET OBJEC.

```
LOOP AT Pnnnn
  WHERE BEGDA LE PC-ENDDA
  AND   ENDDA GE PC-BEGDA.
  WRITE...
ENDLOOP.
```

© SAP AG 2003

- When the **INITIALIZATION** event occurs, you can set default values for the selection screen.
- The fields for the object ID are defined in **DBPCHSEL** include. This is an internal table (PCHOBJID) that must be filled with **APPEND**.
- You can use the **RH-SEL-ONE-OBJID** macro to suppress multiple selection for the Object ID field.
- You can use the **RH-SEL-KEYDATE** macro to restrict the evaluation period to one key day.



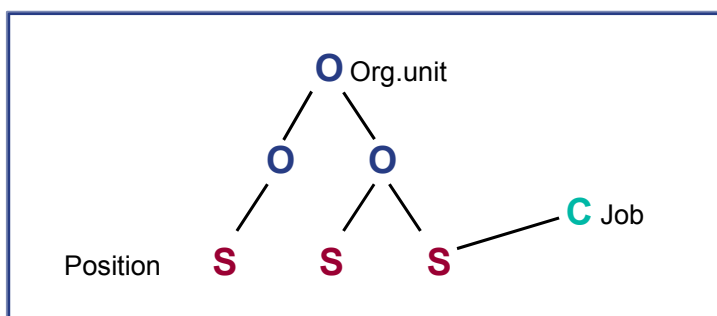
<b>Evaluation path</b>	<input type="text" value="O-S-P"/>	
<b>Status vector</b>	<input type="text" value="12"/>	<b>Status overlap</b>
<b>Display depth</b>	<input type="text" value="3"/>	
<b>Technical depth</b>	<input type="text"/>	<b>Recursion check</b>

© SAP AG 2003

- Enter the required evaluation path in the **Evaluation path** field.
- In the **Status vector** field, enter the status values required by relationship infotype 1001 along the evaluation path so that the appropriate target objects are selected. This parameter enables you to determine the objects irrespective of the status of the relationship infotypes along the evaluation path. For example, enter 12 (without a comma or blank character) to indicate that you only want to display objects whose relationships have status 1 "active" or status 2 "planned".
- The **Status overlap** checkbox is used in conjunction with the status vector field. This enables you to perform a simulation. The results are displayed once all of the relationships have been activated internally. During the simulation, all of the relationships are activated in accordance with the status specified in the Status vector field. Every status value as of position 2 is activated with the status value of position 1. For example, if the status vector is 123 and the status overlay parameter has been set, all relationships in status 2 and 3 are activated with status 1.
- The value entered in the **Display depth** field determines the hierarchical level up to which the structure is displayed. If the value of the field is 3, for example, the system only evaluates and displays the three highest hierarchical levels. In other words, the depth of the root object is 1.
- **Technical depth:** Depth at which a structure is read internally (interactive reporting).
- **Recursion check:** If the system detects recursion (for example, recursive data, recursive evaluation path), the system ends the selection.



<input checked="" type="radio"/> AND relationship	<input type="radio"/> Object filter		
<input type="radio"/> OR relationship	<input checked="" type="radio"/> Branch filter		
<b>CheckObjTy</b>	<b>Root object</b>	<b>Evaluation path</b>	<b>Status vector</b>
<b>S</b>	<b>C</b> 30015502	A007	



© SAP AG 2003

- You can set additional structure conditions that objects must meet. For example, you can evaluate all of the positions along the organizational hierarchy that are also described by one or more specific jobs.
- Read type for structure condition: Objects of the check object type must be accessible from the root object using the evaluation path.
- **AND relationship:**
  - All structure conditions must be fulfilled.
- **OR relationship:**
  - One of the structure conditions must be fulfilled.
- **Object filter:**
  - Irrelevant objects - that is, objects that do not meet the structure conditions - are hidden.
- **Branch filter:**
  - The entire branch below such objects is also hidden.

```
TABLES: OBJEC, GDSTR.
INFOTYPES: nnnn.

INITIALIZATION.
    PCHWEGID = 'SBES'.
. . .
GET OBJEC.
    LOOP AT Pnnnn
        WHERE BEGDA LE PC-ENDDA
        AND ENDDA GE PC-BEGDA.
        WRITE...
    ENDLOOP.
```

© SAP AG 2003

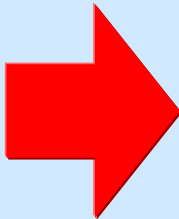
- Some structural evaluations only run with a fixed evaluation path that must not be overwritten on the selection screen.
- If you require a selection screen without structure parameters for such structural evaluations, enter **screen 900** in the report attributes.
- When structural evaluations take place using the sequential selection screen, the evaluation path must be defined in the report.
- The selection screen versions are stored in INCLUDE DBPCHSEL.

```
TABLES: OBJEC, GDSTR.  
INFOTYPES: nnnn.
```

```
. . .
```

```
GET OBJEC.  
  LOOP AT Pnnnn . . .
```

```
ENDLOOP.
```



**Pnnnn** Internal infotype tables

**OBJEC** Object information

**GDSTR** Root of structure

**STRUC** Structure information

© SAP AG 2003

- For each selected object, the internal table **Pnnnn** is filled per infotype with all available infotype records.
- For each selected object, the **OBJEC** structure contains key information from infotype 1000 (plan version, object type, object ID, short text, and long text).
- The **GDSTR** structure contains information on the current hierarchy, such as the root object and the number of objects within the hierarchy. The value of the structure does not change unless the system reaches a new hierarchy root when a structural evaluation is performed for more than one root object.
- For each selected object, the **STRUC** structure contains internal structure information. In particular, it includes information on the relationship used by the system to select the current object along the evaluation path.

- **Sequential evaluations**

```
INFOTYPES: 1001.      "Relationships
TABLES: OBJEC, PADnn.
. . .
GET OBJEC.
    LOOP AT P1001 WHERE RELAT = '0nn'.
        PADnn = P1001-ADATA.
    ENDLOOP.
```

- **Structural evaluations**

```
TABLES: OBJEC, GDSTR, PADnn, STRUC.
.
. . .
GET OBJEC.
    PADnn = STRUC-VADATA.
or:
    IF STRUC-VRELAT = '0nn'.
        PADnn = STRUC-VADATA.
    ENDIF.
```

© SAP AG 2003

- You must ensure that the relationship type is queried (in accordance with table T77AR) before an assignment is made to an additional data structure that is dependent on the relationship type.
- The P1001-ADATA field contains the additional data of the relationships for P1001, that is, the relationships to the current object.
- The STRUC-VADATA field contains the additional data for the current relationship, that is, the relationship that was used to reach the current object.

```
INFOTYPES: 1002.                "Description
TABLES: OBJEC.
DATA: ipt1002 LIKE PT1002 OCCURS 0 WITH HEADER LINE.
. . .

GET OBJEC.
  WRITE: / OBJEC-OTYPE, OBJEC-OBJID.
  LOOP AT P1002.
    RH-GET-TBDAT P1002-INFTY P1002-TABNR ipt1002.
    LOOP AT ipt1002.
      WRITE: / ipt1002-TLINE.
    ENDLOOP.
  ENDLOOP.
```

© SAP AG 2003

- RH-GET-TBDAT is a macro for the logical database PCH. These macros are defined in the include DBPCHCOM. You can use this macro to import the data for an infotype with repetitive structures. For more information, see the relevant pages on **Table Infotypes**.
- The parameters for the macro RH-GET-TBDAT are:
  - Parameter 1 : Infotype
  - Parameter 2 : Reference field
  - Parameter 3 : Table for the PTnnnn structure
- Do not confuse macros for the logical database PCH with macros for the logical database PNP. Note in particular that you cannot use macros for the logical databases PCH and PNP at the same time.

```
INFOTYPES: 1003.                "Department/function
TABLES: OBJEC.

START-OF-SELECTION.
  RH-SET-INDEX-INFTY '1003'.
  RH-CONDITION-LINE ABTEL EQ 'X' SPACE.
  RH-SET-INDEX-INFTY-CONDITION.
  . . .

GET OBJEC.
  WRITE: / OBJEC-OTYPE, OBJEC-OBJID.
  LOOP AT P1003 WHERE ABTEL NE SPACE.
    WRITE: /3 P1003-BEGDA, P1003-ENDDA.
  ENDLOOP.
```

© SAP AG 2003

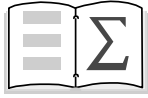
- The call sequence of PCH macros facilitates fast object selection using value conditions of infotype fields (infotype index). You must use this call sequence when objects are selected for sequential evaluations on the basis of whether infotypes exist with specific field values, rather than using the object ID.
- The **RH-SET-INDEX-INFTY** macro specifies the infotype to be indexed.
- The parameters of the **RH-CONDITION-LINE** macro are:
  - Parameter 1 : Field name (for example, ABTEL)
  - Parameter 2 : Condition (for example, EQ, BT)
  - Parameter 3 : Value (for example, "X")
  - Parameter 4 : Value (for Between)
- The **RH-SET-INDEX-INFTY-CONDITION** macro is used to set the condition as an index.
- You can use this method for sequential evaluations, but not for structural evaluations.

```

INFOTYPES: 1001.                "Relationships
DATA: ihobject LIKE HROBJECT OCCURS 10.
. . .
CALL FUNCTION 'RH_READ_INFITY'
EXPORTING
. . .
    INFTY                = '1001'
    BEGDA                = <begdat>
    ENDDA                = <enddat>
TABLES
    INNNN                = P1001
    OBJECTS              = ihobject
EXCEPTIONS
    ALL_INFITY_WITH_SUBTY = 1
    NOTHING_FOUND         = 2
    NO_OBJECTS            = 3
    WRONG_CONDITION       = 4
    OTHERS                = 5.
    
```

© SAP AG 2003

- The RH\_READ\_INFITY function module can be used to read Personnel Planning infotype records (function group RHDB) in programs without the PCH logical database (module pools, reports of other logical databases).
- The function module reads all of the infotype records for a specific infotype or (parameter INFTY = SPACE) all of the infotypes for the specified set of objects.
- Depending on the AUTHORITY and WITH\_STRU\_AUTH parameters, the function module performs an authorization check.



**You are now able to:**

- **Explain the functions of the logical database PCH**
- **Create reports for sequential evaluations of the organizational structure**
- **Create reports for structural evaluations of the organizational structure**
- **Evaluate additional data on relationships**
- **Evaluate table infotypes**





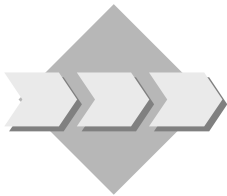
## Unit: The PCH Logical Database

### Topic: Creating a List of Staff Positions



At the end of this exercise, you will be able to:

- Program a structural evaluation
- Select administration infotypes using the PCH logical database
- Understand the search in depth evaluation principle



Your company's Board of Directors requires an overview of the staff positions with the name of the employees holding the positions, and their telephone number.

- 1-1 Write a report that lists the staff positions along the organizational structure, and the names of the employees holding the positions.  
Use the **O-S-P** evaluation path.
  - 1-1-1 Take the personal data (last name, first name) from infotype 0002, and take the telephone number from infotype 0006 subtype 1.
  - 1-1-2 The information on the staff positions is contained in the position in infotype 1003. If a position is marked as a staff position, the value "X" is selected for the **P1003-STABS** field.
  - 1-1-3 Process infotypes 0002 and 0006 of Personnel Administration using a join. For the infotypes, project to the fields to be issued: First name, last name, telephone number, and subtype.
  - 1-1-4 In one row, **only** enter the object ID, the object abbreviation and the object name for the employees holding the positions. In the following row, enter the name and telephone number.
  - 1-1-5 The plan version **01**, object type **O**, object ID **30014999** and evaluation path **O-S-P** should appear as default values.



The PCH logical database displays the object in the following sequence: Organizational unit – Position – Person. The **P1003** internal table is only filled for the object **S** at the time the GET OBJEC event occurs. However, the table is blank again for the following object **P** Person. Therefore, select the appropriate indicator if the position for the person is a staff position.





## Unit: The PCH Logical Database

### Topic: Creating a List of Staff Positions

```
REPORT zhsol010.
TABLES: objec, gdstr.
INFOTYPES: 0002, 0006, 1003.

DATA: stabs LIKE p1003-stabs,
      name LIKE pernr-ename.

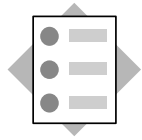
INITIALIZATION.
  pchplvar = '01'.
  pchotype = 'O'.
  pchwegid = 'O-S-P'.
  pchobjid-low = '30014999'.
  append pchobjid.

GET objec.
  IF objec-otype = 'S'.
    CLEAR stabs.
    LOOP AT p1003 WHERE begda LE pc-enddda
          AND enddda GE pc-begda.
      IF p1003-stabs = 'X'.
        stabs = 'X'.
        WRITE : / objec-objid, objec-short, objec-stext.
      ENDIF.
    ENDLOOP.
  ENDIF.
  IF objec-otype = 'P' AND stabs = 'X'.
    PROVIDE vorna nachn FROM p0002
      subty telnr FROM p0006
      BETWEEN pc-begda and pc-enddda
      WHERE p0006-subty = '1'.
    IF p0006_valid = 'X'.
      CONCATENATE p0002-vorna p0002-nachn INTO name
      SEPARATED BY SPACE.
      WRITE: / name, p0006-telnr.
      SKIP.
    ENDIF.
  ENDPROVIDE.
ENDIF.
```



## **Contents:**

- **Enhancement of infotypes**
- **Creating infotypes in Personnel Administration**
- **Creating infotypes in Personnel Planning**



**At the conclusion of this unit, you will be able to:**

- **Add customer-specific fields to standard infotypes**
- **Create a customer-specific infotype in Personnel Administration**
- **Create a customer-specific infotype in Personnel Planning**

Course Overview

Data Structures in  
Personnel Administration

The Logical Databases  
PNP/PNPCE

Join and Projection

Specific Commands

Repetitive Structures and  
List Display

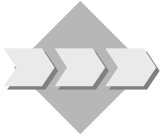
Cluster Database Tables in  
HR

Data Structures in  
Personnel Planning

The Logical Database PCH

 Customer-Specific Infotypes

HR ABAP Features



- Your company has data that cannot be stored in a standard infotype. You want to save this data in customer-specific infotypes.

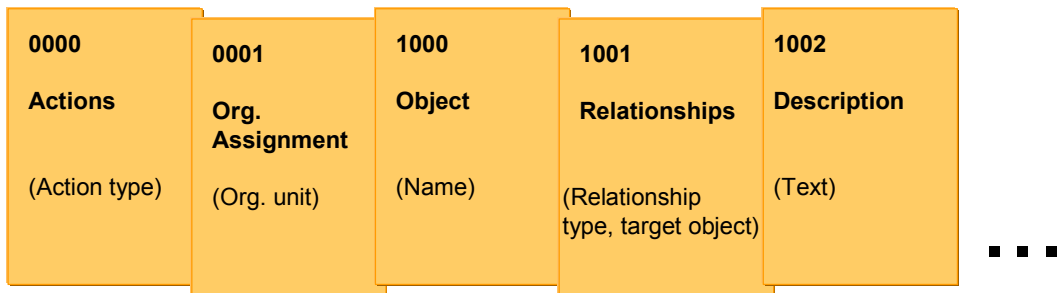


- **"What is an infotype?"**

- **Structural description in the Repository**
- **Database table**
- **Editing program (module pool)**
- **Screens 1000, 2000, 3000**
- **Table entries for control**

Key fields, general fields, PSnnnnn and HRInnnn  
 PAnnnn and HRPnnnn  
 MPnnnn00  
 MPnnnn00 / 1000, 2000, 3000  
 for example, T582A, T582S, T777I, and T777Z

Standard infotypes:

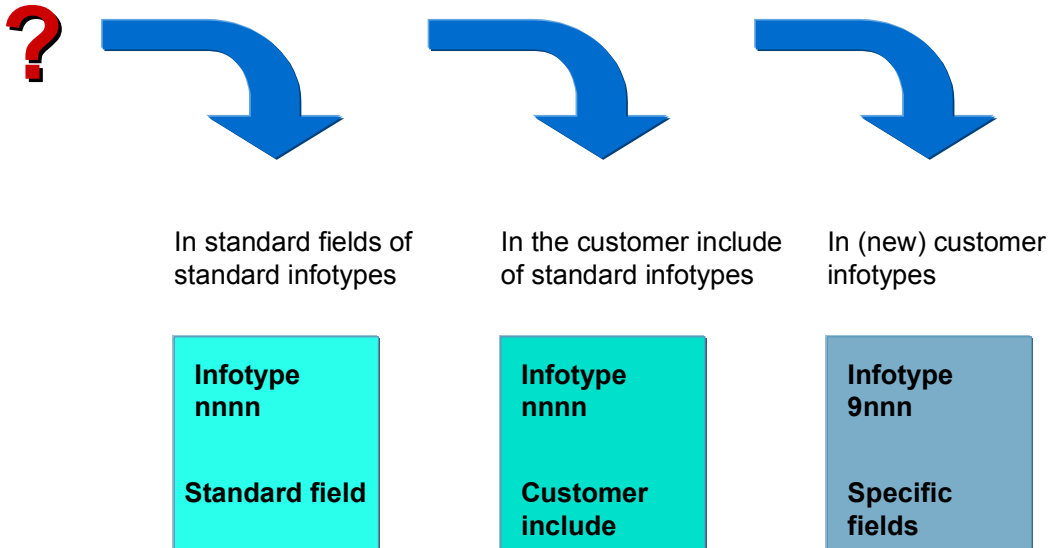


© SAP AG 2003

- Each infotype has an entry in the check table (T777D), and several infotype-dependent components in the Repository.
- In addition to the structures and database tables mentioned in Unit 2 (Data Structures in Personnel Administration), each infotype has a module pool that contains an infotype-specific entry and a list screen with validations, and a dialog module.
- When you create the individual components, you must note that:
  - Specific infotype-dependent objects of infotypes must be edited using the Data Dictionary.
  - You should edit certain components of infotypes only using the specially designed Personnel Planning and Personnel Administration infotype copier.
  - You should make certain infotype-specific settings using infotype Customizing.

If you take account of these notes, you will avoid inconsistencies in infotype-specific objects and table entries. More details on creating and enhancing infotypes are provided in the rest of this unit.

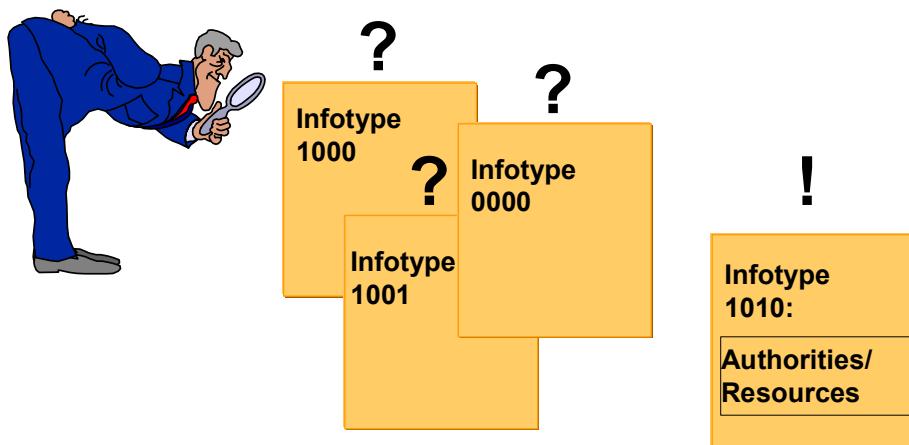
- Where can company-specific information be stored?



© SAP AG 2003

- You have three options for creating customer-specific information in infotypes:
  - You can use a standard infotype with the fields available. For example, you can store the desired information in a specific field of a standard infotype.
  - You can enhance a standard infotype. That is, you can create the fields you require in a customer include.
  - You can create a new infotype with the fields you require.
- Before you enhance standard infotypes or create new infotypes, always check whether the infotypes available in the standard system meet your requirements.

- When can you use standard infotypes?

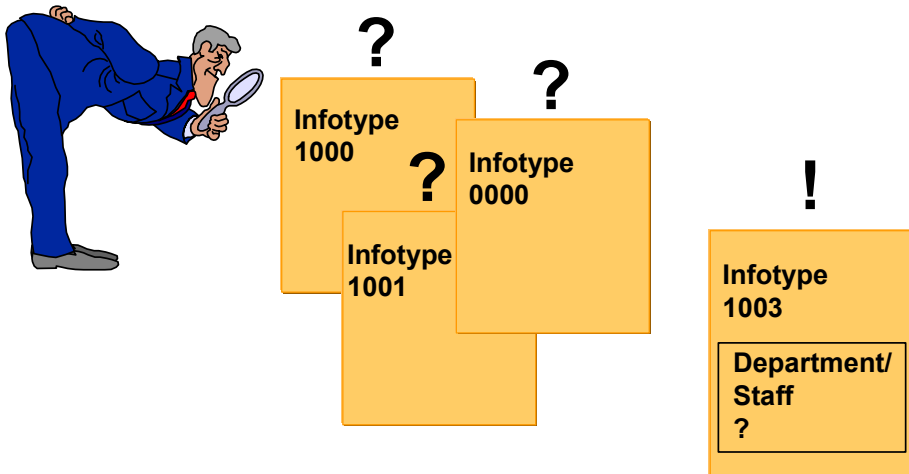


- When a suitable field exists in the standard infotypes

© SAP AG 2003

- Problem: You want to store specific information in the system.
- Procedure: Check all standard infotypes to see if one of the standard infotype fields can be used to store this information.
- Situation: You find an infotype that contains a suitable field.
- Procedure: Use this infotype to store your specific information.
- For example: You would like to store the information “Position entitled to company car”. You can use the standard infotype 1010 (Authorities/Resources) to do this.
- Note: You can retrieve information on the specific fields from the infotype-specific structure Pnnnn (or from the PSnnnn or HRInnnn structures).

- When can you use enhanced standard infotypes?

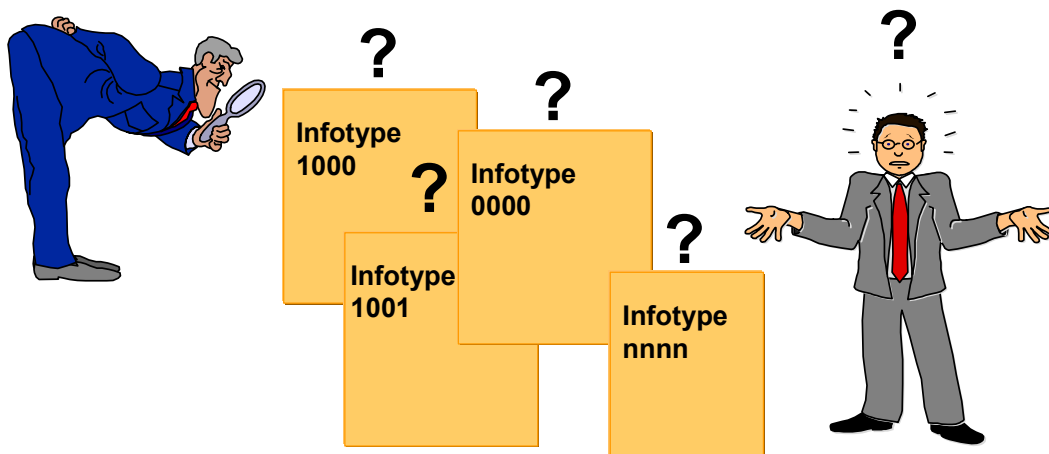


- When a standard infotype meets most of your requirements

© SAP AG 2003

- **Problem:** You want to store specific information in the system.
- **Procedure:** Check all standard infotypes to see if one of the standard infotype fields can be used to store this information.
- **Situation:** You find an infotype that is suitable for most of your data. Additional fields are required, however.
- **Procedure:** Use the customer include structure for this infotype (CI\_Pnnnn) to store specific information by adding the required fields to this structure.
- **Example:** In an organizational unit, you would like to store the information that this organizational unit is managing a project.  
You use the customer include of the standard infotype 1003 (*Department/Staff*), and add the *Project Management* field.
- **Advantage:** The enhanced infotype does not change the infotype already defined in the standard version. In particular, no database conversions are required during upgrades, if SAP simultaneously adds enhancements to this infotype in the standard version (different field sequence possible between Dictionary and Database).  
In addition, evaluations only require minimal change.

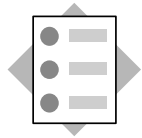
- When can you use customer-specific infotypes?



- When you cannot store the required information in any of the standard infotypes

© SAP AG 2003

- Problem: You want to store specific information in the system.
- Procedure: Check all standard infotypes to see if the fields of one of the standard infotypes can be used to store this information.
- Situation: The information cannot be stored in any of the standard infotype fields.
- Procedure: Use one of the infotype copiers described in the following sections to create a new infotype that meets your specific requirements.



**At the conclusion of this topic, you will be able to:**

- **Add customer-specific fields to a Personnel Administration infotype**

**Contract Elements (infotype 0016)**

**Contractual regulations**

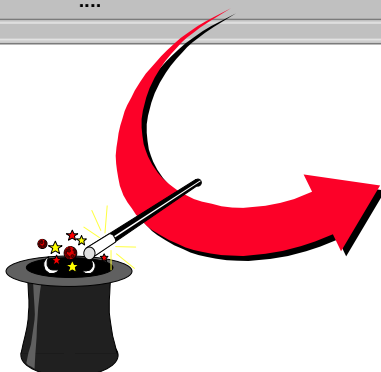
Contract type  Unlimited  
.... ....

**Payment period from beginning of illness**

Continued pay  Days  
Sick pay supplement  Year

**Periods**

Probationary  Months  
.... ....



© SAP AG 2003

**Contract Elements (infotype 0016)**

**Contractual regulations**

Contract type  Unlimited  
.... ....

**Payment period from beginning of illness**

Continued pay  Days  
Sick pay supplement  Year

**Periods**

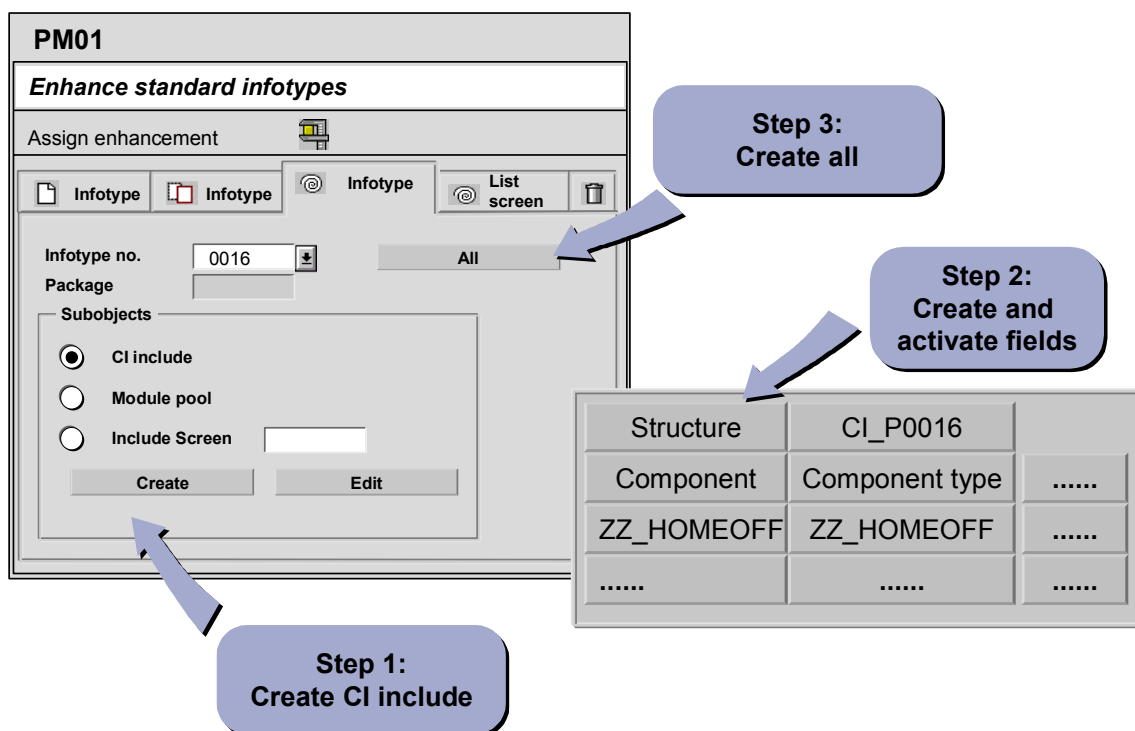
Probationary  Months  
.... ....

**Additional fields**

Customer-specific 1   
Customer-specific 2   
Customer-specific 3



- You can use the enhancement concept to add any number of new fields to a Personnel Administration, Recruitment or Personnel Planning infotype.
- When you add new fields to an infotype, these fields are handled as standard SAP fields in dynamic actions and when creating infotype logs.
- Note that enhancements are not classed as modifications. Adding fields to an infotype in the SAP standard system does not cause any problems during an upgrade.
- The following infotypes are among those not covered by the enhancement concept:
  - Actions infotype (0000)
  - Additional Actions infotype (0302)
  - Time Management infotypes (2nnn)
  - Object infotype (1000)
  - Relationships infotype (1001)



© SAP AG 2003

- Start the infotype copier in Personnel Administration (transaction PM01). The *Create Infotype* screen appears. Select *Expand Infotype*. In the *Infotype Number* field, enter the four-digit number of the infotype you would like to enhance. When you enter the infotype number, remember to enter leading zeros as well.
- In the *Subobjects* groupbox, select *CI include*, and choose *Create*. The screen *Dictionary: Initial Screen* appears.
- Create the CI include and choose *Activate*. If you would like to add completely new fields, you must also create data elements. Go back to the *Enhance Infotypes* screen.
- Select *Create All*.
- Result: You have added additional fields to the standard single screen of an infotype.



- **CI\_Include**      **C\_Pnnnn**      **Customer-specific data fields**
- **Module pool**      **ZPnnnn00**
- **Include screen**      **ZPnnnn00 0200**
- **Includes**      **ZPnnnn10**      **Declaration of joint data objects**  
                          **ZPnnnn20**      **PBO module for screens**  
                          **ZPnnnn30**      **PAI module for screens**  
                          **ZPnnnn40**      **Subroutines**

© SAP AG 2003

- The include screen is assigned to the standard screen. To do this, use the *Assign enhancement* button. The entries in table T582C ensure that the screen underneath the standard fields is displayed.
- Calling a subscreen for the processing times PBO and PAI: CALL SUBSCREEN SUBSCREEN\_T582C cause the customer enhancement to be taken into account in the module pool of the standard screen.



**At the conclusion of this topic, you will be able to:**

- **Create a customer-specific infotype in Personnel Administration**

**PM01**

**Create Infotype**

Infotype characteristics    Technical attributes   

☐ Infotype

Infotype no.    9nnn    All

Package   

☐ Employee infotype    ☐ Applicant infotype    ☐ Both

Subobjects

☒ PS structure

☐ Module pool

☐ User screen   

☐ Interface

**1. Create and activate PS9nnn structure**



**Step 2:**

© SAP AG 2003

- Step 1: Call the Personnel Administration infotype copier (transaction PM01).
- You must store the infotype-specific fields for your infotype in the system. To do this, create the structure PS9nnn in the Dictionary.
  - Save and activate the structure.
  - You may need new data elements and domains for your structure.

**PM01**

**Create Infotype**

Infotype characteristics Technical attributes

Infotype

Infotype no. 9nnn

Package

☐ Employee infotype ☐ Applicant infotype ☐ Both

Subobjects

☒ PS structure

☐ Module pool

☐ Screen

☐ User Interface

Step 1:

2. Create and activate all objects

■ Step 2: Once you have created the PS9nnn structure, choose *Create All* in transaction PM01.

■ You now create a new infotype.

- When you do this, the infotype copier automatically creates the following objects:
  - Database table PA9nnn
  - Structure P9nnn
  - Module pool: MP9nnn00
  - 3 screens (1000, 2000, 3000) for the module pool
  - Interface status
  - Dialog module (RP\_9nnn)
- When you create new infotypes, the following table entries are also automatically created:
  - T777D Infotype check table
  - TDCD Dialog modules

## Creating Personnel Administration Infotypes - Step 3

SAP

<b>Infotype</b>		<b>9001</b>	<b>Company carpark</b>	
<b>General attributes</b>				
Time constraint	<b>2</b>	<input type="checkbox"/> Subtype obligatory	<input type="checkbox"/> Accntng/log.data	
Time cnstr. tab.		Subtype table	<input checked="" type="checkbox"/> Text allowed	
Maint.aft.leave	<input type="checkbox"/>	Subty. text tab.	<input type="checkbox"/> Copy infotype	
<input checked="" type="checkbox"/> Access auth.		Subtype field	<input type="checkbox"/> Propose infotype	
<b>Display and selection</b>				
Select w/start	<b>3</b>	Create w/o strt	<b>1</b>	Sort sequence
Select w/ end	<b>5</b>	Create w/o end	<b>1</b>	Screen header
Select.w/o date	<b>2</b>	<input type="checkbox"/> List time per.		<input checked="" type="checkbox"/> Choose data
<b>Retroactive accounting trigger</b>				
<input type="checkbox"/> Before ERA date		<input type="checkbox"/> Entry of RA limit time		Past entry all.
Retr.acct.payr.	<input type="checkbox"/>	Retr.acct.PDC	<input type="checkbox"/>	No org.assign.
<b>Technical data</b>				
Single screen	<b>2000</b>	Dialog module	<b>RP_9001</b>	Applicant infotypes
List screen	<b>3000</b>	Structure	<b>P9001</b>	Applicant DBTab
<input type="checkbox"/> List entry		Database table	<b>PA9001</b>	

- When you create an infotype, the system does not automatically create the table entries that describe the infotype's characteristics (tables T582A and T582S). You must therefore create the corresponding entries manually in the tables. You use the view V\_T582A to maintain these tables. This view appears after you carry out the *Create All* step.
- You can maintain these tables at any time in transaction PM01 on the *Create infotype* screen. To do this, select *Infotype characteristics*.
- You can create new entries by:
  - Selecting *New Entries*, or
  - Copying an entry for an infotype with similar characteristics

- **Structures P9nnn and PS9nnn**
- **Database table PA9nnn**
- **Module pool MP9nnn00 with includes**
  - MP9nnn10 (data declaration)**
  - MP9nnn20 (PBO modules for screens)**
  - MP9nnn30 (PAI modules for screens)**
  - MP9nnn40 (Subroutines)**
- **Screens 1000, 2000, and 3000 for module pool**
- **GUI status with standard icons**
- **Dialog module RP\_9nnn**
- **Entries in T777D check table infotypes and TDCD dialog modules**

© SAP AG 2003

- Select *Create All* in transaction PM01 to create the objects outlined above.
- In the screens, you often have to modify the *Layout* and *Flow logic* components.
- You usually do not have to make any further modifications to the GUI status.

```

PROGRAM MP9nnn00 MESSAGE-ID RP.

TABLES: P9nnn.
* the following tables are filled globally:
* T001P, T500P
* they can be made available with a TABLES statement

FIELD-SYMBOLS: <PNNNN> STRUCTURE P9nnn
                DEFAULT P9nnn.

DATA: PSAVE LIKE P9nnn.
    
```

© SAP AG 2003

- Global data is created in the MP9nnn10 include. These are the declarations for the tables and work fields used.

```
MODULE P9nnn OUTPUT.  
  IF PSYST-NSELC EQ YES.  
    * read text fields etc.; do this whenever the screen is  
    * shown for the first time:  
    *   PERFORM RExxxx.  
    IF PSYST-IINIT = YES AND PSYST-IOPER = INSERT.  
    * generate default values; do this the very first time  
    * on insert only:  
    *   PERFORM GET_DEFAULT.  
    ENDIF.  
  ENDIF.  
ENDMODULE.
```

© SAP AG 2003

- The MP9nnn20 include contains the modules that are called when the PBO event occurs. To allow the program to be structured better in subroutines, the processing of the data is defined in the include MP9nnn40.
- To control processing, it is useful to query specific values of the **PSYST** structure.
- The **NSELC** indicator has the value YES (= 1), if a record is displayed on screen for the first time. It should be queried, so that it is only read once when the PBP text for keys is run repeatedly.
- The **IINIT** indicator has the value YES, if a record is displayed for the first time. It is used to carry out special processing actions of the PBO for this record, for example, inserting default values when adding elements.
- The only difference between these indicators is that when you go to the overview when processing a record, IINIT is still on NO (= 0) after you go back to processing. This prevents default values that are changed by the user from being renewed. If a user overwrites the default values during creation, and goes to the overview during entry, the variable IINIT is at NO when you return to the entry screen. The default values are not to be redetermined. However, the NSELC variable has the value YES as the system must read the texts again.



```
PROCESS BEFORE OUTPUT.  
*      General infotype-independent processing  
MODULE BEFORE_OUTPUT.  
  CALL SUBSCREEN subscreen_empl  
    INCLUDING empl_prog empl_dynnr.  
  CALL SUBSCREEN subscreen_header  
    INCLUDING header_prog header_dynnr.  
*      Infotype-specific processing  
MODULE P9nnn.  
*  
MODULE HIDDEN_DATA.
```

© SAP AG 2003

- You can carry out infotype-specific initializations in the **P9nnn** PBO module. Here, you can, for example, fill the screen fields that are stored in the **Qnnnn** (SAP) and **Znnnn** (customer) structures.
- Do not change the **BEFORE\_OUTPUT** and **HIDDEN\_DATA** PBO modules.

```
PROCESS AFTER INPUT.  
*   Exit processing.  
    MODULE EXIT AT EXIT-COMMAND.  
*   Processing after input  
*   Note when entry has been made: All screen fields that  
*   are ready for input must be listed here  
  
CHAIN.  
    FIELD P9nnn-BEGDA.  
    FIELD P9nnn-ENDDA.  
    FIELD P9nnn-field1,...  
    MODULE INPUT_STATUS ON CHAIN-REQUEST.  
ENDCHAIN.
```

© SAP AG 2003

- The **INPUT\_STATUS** PAI module must be executed if the user makes an entry in a screen field. Therefore, all entry fields must be listed in the chain.

```

*   Function code editing before entry checks
MODULE PRE_INPUT_CHECKS.
*   Entry checks:
*   Insert check modules here:
    FIELD P9nnn-field1 ON INPUT MODULE CHECK_FIELD1.
*   Processing of function code entry: ALL other fields
*   that appear on the screen must be listed here, even
*   the only output fields
CHAIN.
    FIELD P9001-BEGDA.
    FIELD P9001-ENDDA.
    FIELD RP50M-SPRTX.
    FIELD P9nnn-field1,...
    MODULE POST_INPUT_CHECKS.
ENDCHAIN.
    
```

© SAP AG 2003

- The function code is processed before the entry check in the **PRE\_INPUT\_CHECKS** PAI module. The system exits processing of the current individual screen if, for example, you select *Exit*.
- Once you have processed the **PRE\_INPUT\_CHECKS** module, you can make your own entry checks or call your own PAI modules. In the above example, an entry in the *P9nnn-field1* field is checked by the **CHECK\_FIELD1** module.
- The entry checks must be completed up to the **POST\_INPUT\_CHECKS** PAI module. It is not possible to change field contents after subsequent processing.
- The function code is processed after the entry check in the **POST\_INPUT\_CHECKS** PAI module. General entry checks are also carried out here. For the general entry checks, the system checks for example, whether the start date of the infotype record is before the record's end date.
- All fields that appear on the single screen must be listed in the chain. Fields that are only displayed, for example, long texts, are also to be listed here.

Screen field is ready-for-input for the function	Hexadecimal constant for modification group 1
<i>Display</i>	<b>001</b>
<i>Change</i>	<b>002</b>
<i>Add and copy</i>	<b>004</b>
<i>Delete</i>	<b>008</b>
<i>Lock/Unlock</i>	<b>010</b>

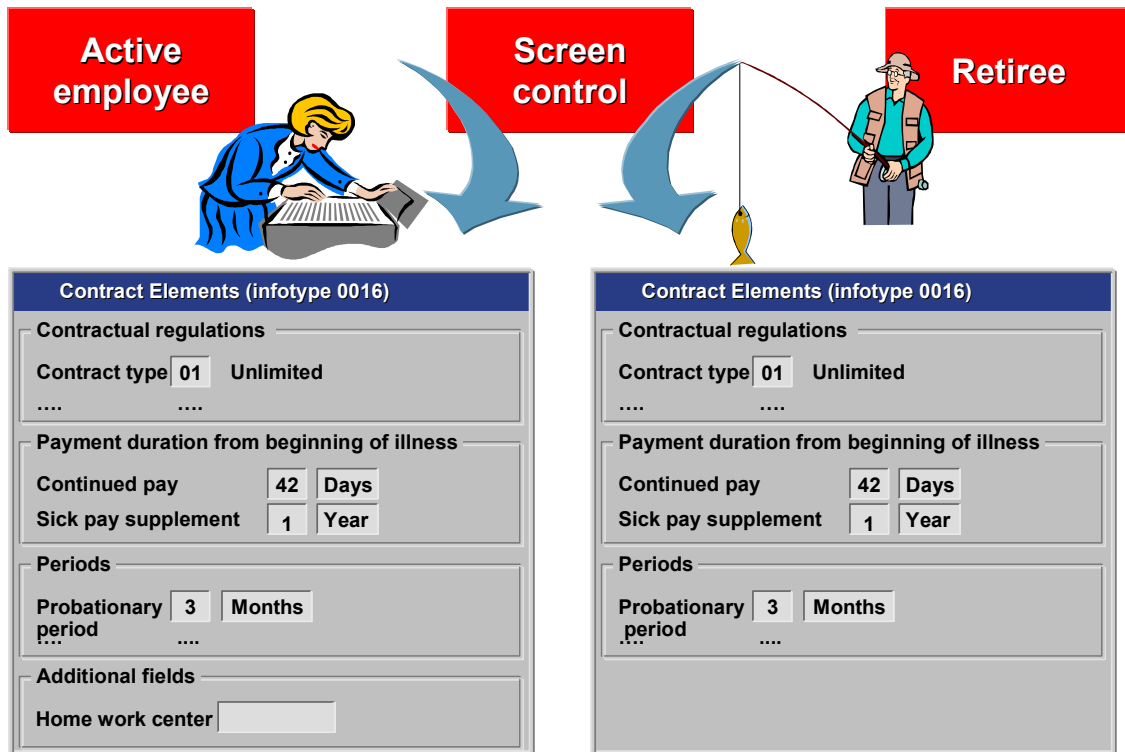
© SAP AG 2003

- You determine the attributes of the individual screen fields when you create the single screens and list screens with the ABAP Screen Painter. However, the same screen is always used for different functions such as creating, displaying, maintaining, and deleting infotype records. Therefore, when you maintain the screen, you cannot specify whether or not a screen field is to be ready for input. Depending on the employee's organizational data, certain screen fields may need to be hidden.
- Some attributes can only be specified at runtime, and cannot be specified in general. The screens' layout is therefore to change, depending on the function a user has chosen, or on the data to be edited.
- Controlling whether screen fields are ready for input or hidden:
- You can use the *Modification group 1* value to determine whether screen fields are ready for input, depending on the function to be carried out. You can also hide specific screen fields.
- The value of *Modification group 1* must be maintained for all input fields.
- A screen field is not ready for input if you have not maintained this entry.

Screen field is hidden for the function	Hexadecimal constant for modification group 1
<i>Delimit in list screen</i>	<b>200</b>
<i>Display in list screen and change in list screen</i>	<b>400</b>
<i>Add and copy</i>	<b>800</b>

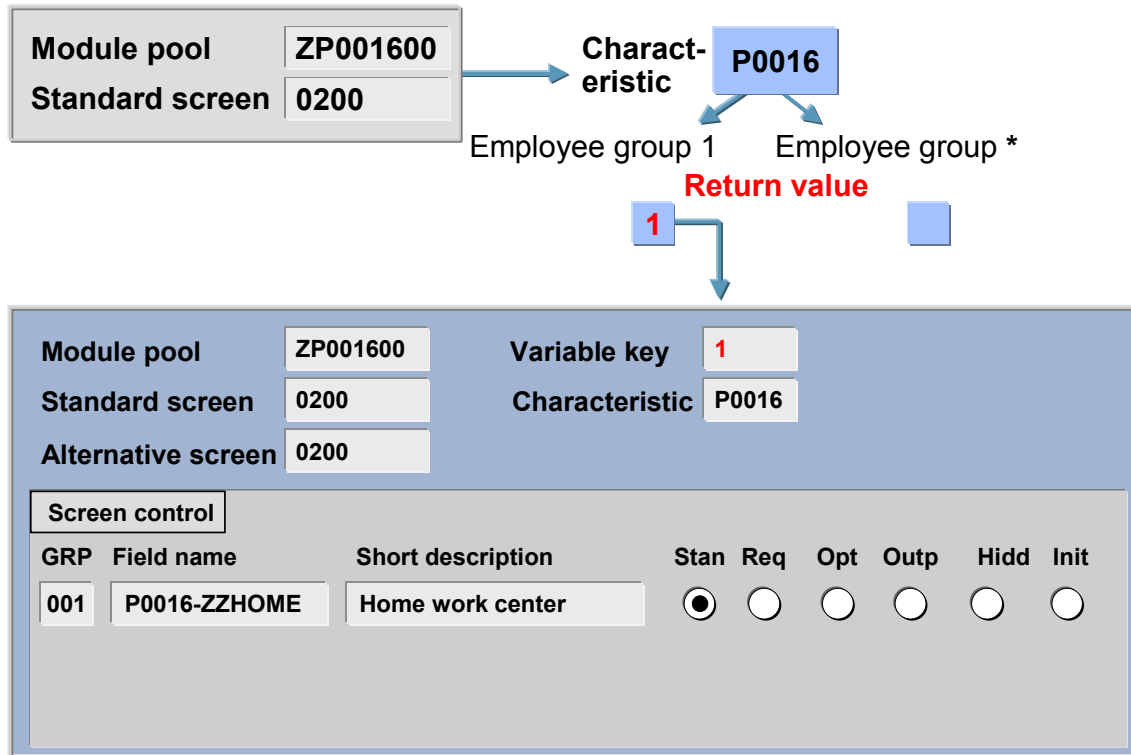
© SAP AG 2003

- The value of *Modification group 1* is interpreted bit-by-bit. Several constants can be combined with each other. For this reason, the values are added. Note that the *Modification group 1* value must be maintained in hexadecimal form.
- For example:  
You would like a screen field to be ready for input for the *Add* and *Change* functions. You therefore maintain the value **006** in *Modification group 1*. You would like one screen field to be ready for input for all functions. Here, you maintain the value **00F** in *Modification group 1*.
- The **BEGDA** and **ENDDA** fields are usually ready for input for all actions except for the display of records. You therefore assign the value **00E** to the *Modification group 1* attribute of these fields.
- *Modification group 1* has the value **800** for the *AEDTM* and *UNAME* fields. This means that these fields are hidden when you add a record.



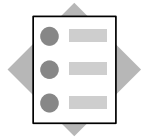
© SAP AG 2003

- Together with control data, you can use screen control to:
  - Replace the standard screen with an alternative screen
  - Control whether specific screen fields are ready for input
  - Hide specific screen fields
- In general, screen control can be carried out based on organizational data for the employee or based on the infotype record's subtype. To this end, maintain the value of *Modification group 3* in the ABAP Screen Painter for the corresponding screen fields.
- Each screen field is assigned a value between **001** and **050** in *Modification group 3*. Use the same value for screen fields that are to be modified in the same way. The same value is used for an input/output field as for the corresponding keyword and any long text that is displayed. Assign screen fields that are not modified using T588M to the value *SPACE* in *Modification group 3*.



© SAP AG 2003

- In table T588M, you then determine which alternative screens you would like to use, if any, and how the individual screen fields are to be modified.
- If you require a modified screen, but do not find any entry in this table, create a new basic entry for the screen and make sure that the *Variable key* field is empty.
- If the screen modification is to apply for the infotype in general, you do not require any characteristic or any other entries in this table.
- As is the case in this example, the screen modification can be based on the employee's organizational assignment. The modification is determined by the Pnnnn characteristic (nnnn = infotype number). If you do not enter a return value in the characteristic's decision tree, the entry for which the *Variable key* field is empty, that is the basic entry (basic screen), is automatically valid. If no characteristic is available for the corresponding infotype, you must create one.
- Create a new entry with the return value in the variable key for each return value for the characteristic. If necessary, assign the screen number of an alternative screen. The screen modification in the detail view is now also valid for the alternative screen.



**At the conclusion of this topic, you will be able to:**

- **Add customer-specific fields to a Personnel Planning infotype**



**PPCI**

**Enhance Infotype**

Infotype: 1003  
Infotype name: \_\_\_\_\_

Single screen:  
☒ CI include All  
☐ Module pool  
☐ Include screen 200

Create Change

**Step 1: Create CI include**

Structure	CI_P1003	
Component	Component type	.....
ZZ_PROJECT	ZZ_PROJECT	.....
.....	.....	.....

**Step 2: Create and activate fields**

**Step 3: Create All**

© SAP AG 2003

- Start the Personnel Planning infotype copier (transaction PPCI). In the *Infotype* field, enter the four-digit number of the infotype you would like to enhance. Select *Extend*.
- The *Enhance Infotype* screen appears. Select *Create All*. The *Dictionary: Initial Screen* appears.
- Create the include and choose *Activate*. If you would like to add completely new fields, you must also create data elements. Go back to the *Enhance Infotype* screen.
- Result: You have added additional fields to the standard single screen of an infotype.
- **Note:**
  - It is not possible to enhance the infotypes 1000 (*Object*) or 1001 (*Relationships*).
  - Before you enhance infotypes using SE16, check whether the MAINT field is filled in table T777I. If this is the case, the infotype cannot be directly maintained and it is not possible to add an enhancement.



**At the conclusion of this topic, you will be able to:**

- **Create a customer-specific infotype for Personnel Planning**

**ABAP Dictionary: Initial Screen**

• Database table	<input type="text"/>
• View	<input type="text"/>
• Datatype	<input type="text" value="HRI9nnn"/>
• Type group	<input type="text"/>
• Domain	<input type="text"/>
• Search help	<input type="text"/>
• Lock object	<input type="text"/>

Display Change Create

1. Create and activate structure

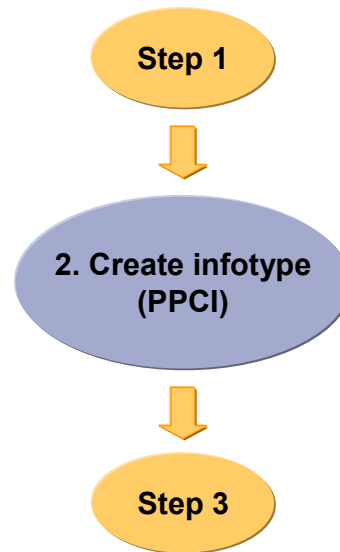
Step 2

Step 3

© SAP AG 2003

## Step 1:

- You must store the infotype-specific fields for your infotype in the system. To do this, create the HRI9nnn structure in the Dictionary.
  - Note that when you do this, you are creating a *Data type* and not a *Database table*.
  - Save and activate the structure.
  - You may need new data elements and domains for your structure.
- If your infotype's data has a repetitive structure, that is, you wish to create a table infotype, you must create the PT9nnn structure for the table part of your infotype. If your table infotype is not composed solely of table entries, but also of individual fields, you must also create the HRI9nnn structure.



© SAP AG 2003

## ■ Step 2:

- Call the Personnel Planning infotype copier (transaction PPCI).
- Enter the four-digit infotype number 9nnn in line with the structure you created and select *Create*.
- When you create the infotype, you can choose between field infotypes and table infotypes. You can also determine whether the infotype is language- and/or country-specific.
- Create a new infotype. When you do this, the infotype copier automatically creates the following objects:
  - Database table HRP9nnn
  - Module pool MP9nnn00
  - Table index
  - 3 screens (1000, 2000, 3000) for the module pool
  - The structure P9nnn
  - The CUA interface- A dialog module (RH\_INFOTYP\_9nnn)
- When you create new infotypes, the following table entries are also automatically maintained:
  - T777D Infotype check type
  - T77ID Enhancements for T777D
  - T778T Infotypes
  - T777T Infotype texts
  - TDCT Dialog modules

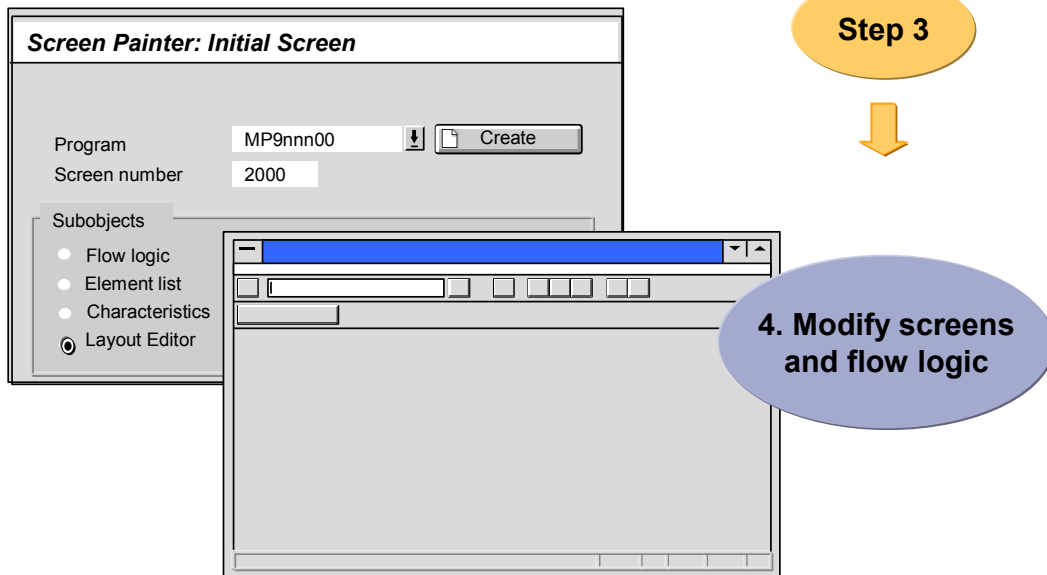
Step 2

3. Maintain tables

Step 4

© SAP AG 2003

- After you create the infotype, entries must be made in the following tables:
  - T777I Infotypes per object type
  - T777Z Time constraint of infotypes
  - T77CD Customer-specific infotype settings
  - T77NI Country-specific infotypes
- Step 3: Select *Check*.
- On the following screen, position the cursor on the table you would like to maintain and select *Change*.
- You can also open these tables in Customizing (path: *Personnel Management* → *Organizational Management* → *Basic Settings* → *Data Model Enhancement* → *Infotype Maintenance*).
- **Note:**  
 You must create the create screen 7000 if you wish to maintain the new infotype using the transaction PPOME. To do this, select *Infotype* → *Create subscreen*. The entry in table T77ID is also enhanced. To make an assignment to a tab, open the IMG (path: *Personnel Management* → *Organizational Management* → *Hierarchy Framework*), and maintain the entries *Adjust Tab Pages in Detail Area* and *Integrate New Infotype* for the scenario OME0.

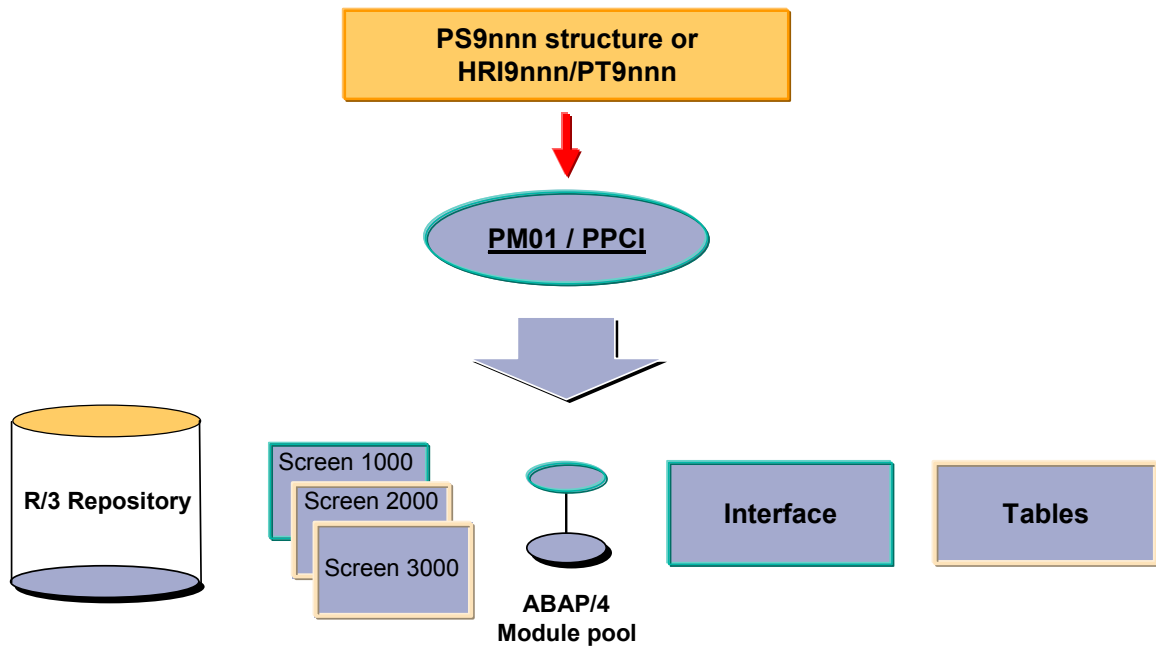


© SAP AG 2003

## ■ Step 4:

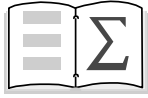
- Modify the new screens to suit your needs.
- To do this, edit the layout, the flow logic, and the field characteristics (modification groups).
- You can define infotype-specific check modules in flow logic in include MP9nnn20.
- The default value 036 is set for the first modification group. When you start the report RHGROUP1 using this value, you establish for which functions (for example, create, change, and insert) the corresponding field is ready for input for this value. If you want to use another entry control, use the report RHGROUP0. This calculates the value for the first modification group, based on the function codes you select.

- **Creating New Infotypes - Overview**



© SAP AG 2003

- To sum up: There are three steps for creating customer-specific infotypes:
  - Definition of the info-specific fields in the structures PS9nnn and HRI9nnn in the Dictionary
  - Creation of the central objects of the infotypes using transactions PM01 and PPCI (Personnel Administration and Personnel Planning infotype copiers)
  - Customer-specific modifications and controls



**You are now able to:**

- **Add customer-specific fields to standard infotypes**
- **Create a customer-specific infotype in Personnel Administration**
- **Create a customer-specific infotype in Personnel Planning**





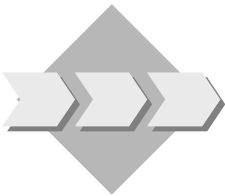
## Unit: Customer-Specific Infotypes

### Topic: Creating an Administration Infotype



At the end of this exercise, you will be able to:

- Create and activate the PS9nnn structure
- Maintain the T582A table
- Modify the infotype single screen



Your company's Personnel Department would like to keep a record of home work centers for individual employees.

- 1-1 Create the *Home work center* (##) infotype for Personnel Administration. Use infotype number **95##**. (## = group number)
  - 1-1-1 Enter the following fields for the infotype:
    - Field 1: **COMP** *Fulltime*, new data element: **ZCOMP##**, domain: CHAR1\_X.  
This field indicates whether the home work center is fulltime.
    - Field 2: **EQUI** *Equipment*, new data element: **ZEQUI##**, domain: CHAR8.  
Use this field to record IDs for possible equipment for the home work center.
  - 1-1-2 Maintain the infotype characteristics (T582A). You can use the characteristics of infotype 0002 as a template.
  - 1-1-3 Convert the *Fulltime* field to a checkbox using the Screen Painter.
  - 1-1-4 Test your infotype using personnel master data maintenance (transaction PA30).
- 1-2 (optional) Enhancement of flow logic of the infotype for the PBO event using a check table for the *Equipment* field.
  - 1-2-1 In your structure **PS95##**, create the foreign key relationship for the *EQUI* field. Enter the **ZZEQUI** table as the check table.
  - 1-2-2 Change the screen for the detail screen for your new infotype, so that the corresponding descriptive text appears next to the *Equipment* field.
  - 1-2-3 Test your development in HR master data maintenance.

- 1-3 (optional) Enhancement of flow logic of the infotype for the PAI event.
  - 1-3-1 Enhance the flow logic of the detail screen for your new infotype so that it is only possible to activate the *Fulltime* field if the *Part-time worker* field is not selected in infotype 0007 *Planned Working Time*. If this field is selected, error message 855 is to be issued from the *RP* message class.
  - 1-3-2 Test your development in personnel master data maintenance.

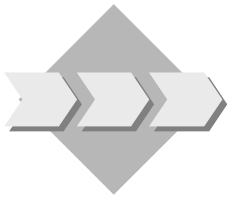


## Unit: Customer-Specific Infotypes

### Topic: Creating a Table Infotype



- Create and activate the required structures for Personnel Planning infotypes.
- Become familiar with transaction PPCI.
- Maintain tables T777I and T777Z.



Data on the size and equipment of offices is required for office planning. Set up the system so it is possible to store this data.

2-1 Create the *Office equipment ##* infotype as a table infotype for Organizational Management. Use infotype number 96##. (## = group number)

2-1-1 Insert the following fields in the non-table part of the infotype:

Field 1: **AREA** *Area*, new data element: ZAREA##, datatype DEC, length 8, 2 decimal places.  
This field stores the area of an office.

Field 2: **NWORK** *Number of work centers*, new data element: ZNWORK##, data type DEC, length 5, no decimal places.  
This field records the maximum number of work centers in an office.

2-1-2 Insert the following fields in the table part of the infotype:

Field 1: **SNWRK** *Work center number*, new data element: ZSNWRK##, data type DEC, length 5, no decimal places.  
This field stores the sequence number of the work centers in an office.

Field 2: **EQUI** *Equipment*, data element: ZEQUI## from the previous exercise.  
This field contains the identifier for the equipment for a work center.

2-1-3 Extend the data model of Organizational Management so that you can maintain your new infotype for the object type **O** *Organizational Unit*.

2-1-4 Test your new infotype in expert mode for the organizational unit ## *Controlling*.





## Unit: Customer-Specific Infotypes

### Topic: Creating an Administration Infotype

#### Note:

- Create all development objects as local objects.
- You must activate the objects before you can test your developments.

1-1 Call transaction **PM01** and enter **95##** in the *Infotype number* field. (## = group number)

1-1-1 Select *PS Structure* and choose *Create*. This automatically takes you to the Data Dictionary structure maintenance screen. In the *Short Description* field, enter the name of your infotype, for example, *Home work center ##*. Enter the field name in the *Components* column, and enter the data element name in the *Component type* column.

Field 1: Name COMP, data element: ZCOMP##

Field 2: Name EQUI, data element: ZEQUI##

#### To create the data element:

Double-click the data element name to go to create mode for component types. Select *Data element* in the dialog box that appears.

Enter relevant short descriptions for your data elements, enter the name of the referenced domain - CHAR1\_X or CHAR8 –, maintain the field labels and the documentation, if relevant.

Save and activate both data elements.

Save and activate the structure.

Select F3 to go back to the *Create Infotype* transaction and select *Create All*.

1-1-2 Maintain the infotype characteristics (T582A) by copying the characteristics from infotype 0002. In the copied entry, correct the name of the infotype and the time constraint. 2 is the correct time constraint for your infotype.

Select F3 to go back to the *Create Infotype* transaction.

1-1-3 To convert the *Fulltime* field to a checkbox: In the *Create Infotype* transaction, select *Screen* under *Subobjects* and enter the number *2000*. Select *Edit*. This takes you to the *Screen Painter*. Select the Graphical layout editor from *Utilities* → *Settings*.

Select *Change* and the *Layout Editor* subobject. Select the field and the field text of the COMP field, and choose **Edit → Convert → Checkbox → Left button**. Save and activate the screen.

1-1-4 To test the new infotype: Call HR data maintenance (transaction PA30), enter an existing personnel number (for example, 1000), enter the number of your infotype (95##), and select *Create*. Enter any data, save, and test the different functions in personnel master data maintenance, such as *Change*, *Copy*, *Delimit*, *Delete*.

1-2 (optional) Enhancement of flow logic of the infotype for the PBO event.

1-2-1 Call transaction **PM01** and enter 95## in the *Infotype number* field (## = group number). Select *PS Structure* and choose *Edit*. This automatically takes you to the Data Dictionary structure maintenance screen. Select the line with the *EQUI* field, choose the *Entry help/check* tab, and select the *Foreign Keys* icon.

The following foreign key relationship is required:

<u>Check table</u>	<u>Check table field</u>	<u>Foreign key table</u>	<u>Foreign key field</u>
ZZEQUI	MANDT	SYST	MANDT
ZZEQUI	EQUI	PS95##	EQUI

1-2-2 Change the screen for the detail screen for your new infotype, so that the corresponding descriptive text appears next to the *Equipment* field.

a) In the include **MP95##10**, insert the following ABAP line:

```
tables zzequit.
```

b) Insert the following statement as the first line of the PBO module **P95##**:

```
select single * from zzequit where equi = p95##-equi
                        and langu = sy-langu
```

c) In the layout of screen 2000, enter the field ZZEQUIT-EQUITXT. To do this, call the *Screen Painter* for the screen, go to the *Layout Editor* and select → **Goto → Secondary Window → Dict/Program Fields**. Enter the table name ZZEQUIT, select *Get from Dictionary*, choose the line with the field EQUITXT and select *Enter* to transfer the field *Without Text*. Now use the mouse to position the field on the screen.

1-3 (optional) Enhancement of flow logic of the infotype for the PAI event.

1-3-1 Enhance the flow logic of the detail screen for your new infotype so that it is only possible to activate the *Fulltime* field if the *Part-time worker* field is not selected in infotype 0007 *Planned Working Time*.

- a) In the flow logic for screen 2000 for the PAI event after the PRE\_INPUT\_CHECKS module, enter the following line:

```
*-----*
*   Entry checks                               *
*-----*
*   Insert check modules here:
field p9500-comp on input module p95##_checks.
```

- b) Double-click p95##\_checks to create this new PAI module. Create it using the following ABAP lines in include **MP95##30**:

```
MODULE P95##_checks INPUT.
  SELECT SINGLE * FROM pa0007
    where pernr EQ p95##-pernr
    and   begda le p95##-endda
    and   endda ge p95##-begda.
  IF NOT pa0007-teilk IS INITIAL.
    MESSAGE e855(rp) .
  ENDIF.
ENDMODULE.
```



## Unit: Customer-Specific Infotypes

### Topic: Creating a Table Infotype

2-1 Create the *Office equipment (##)* infotype as a table infotype for Organizational Management. Use infotype number 96##. (## = group number)

2-1-1 To create the structure for the non-table part of the infotype:  
Call the *Data Dictionary*, and create a new structure called **HRI96##**. Use the following field and data element name:

AREA	Data element ZAREA## Data type DEC, length 8, 2 decimal places
NWORK	Data element ZNWORK## Data type DEC, length 5, no decimal places

To create the data element:

Double-click the data element name to go to create mode for component types.  
Select *Data element* in the dialog box that appears.

Enter relevant short descriptions and field labels for your data elements. Select *Built-in type* for the data type information and use the information provided above.

Save and activate both data elements.

Save and activate the structure.

2-1-2 To create the structure for the table part of the infotype:  
Call the *Data Dictionary*, and create a structure called **PT96##**. Use the following field and data element name:

SNWRK	Data element ZSNWRK## Data type DEC, length 5, no decimal places
EQUI	You created the data element ZEQUI## in the exercise on creating an administration infotype.

Create the data element ZSNWRK as described in 1-1-1.

Save and activate the new data element.

Save and activate the structure.

Call transaction PPCI, enter the infotype number 96## and the description "Office equipment ##". Select *Create*. On the next screen, select the *Table infotype* option and then *Create* in the *Infotype* type.



- 2-1-3 Carry out enhancements in the Organizational Management data model so that you can maintain your new infotype for the *Organizational unit* object type **O**:

In transaction PPCI, select *Check*. In the dialog box that appears, position the cursor on line *T777I: ...* in the *Table Entries* section and select *Change*. Go to table maintenance, and position the cursor on infotype 96##. Select the entry and choose *Time constraint* in the dialog structure. Create a new entry for *Time constraint* with the values:

Object type: O  
Infotype: 96##  
Subtype:  
Time constraint: 2

Maintain a new entry for *Infotypes per object type* with the values:

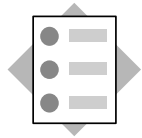
Object type: O  
Infotype: 96##

- 2-1-4 Test the new infotype in expert mode in Organizational Management. Call transaction PP01, or in the SAP Menu, select → *Personnel* → *Organizational Management* → *Expert Mode* → *General*. Select the *Current Plan* plan version and the *Organizational Unit* object type. Call the possible entries help for the *Object ID* field and select the *Other search help* icon. Select the search term search help and enter ## **Cont**. In the infotype table, enter your infotype "Office equipment ##", and create an infotype record. Test other functions of this transaction.



### **Contents:**

- **Calling Routines Using Dynamic Actions**
- **Selection Reports for Fast Entry**
- **Calling Subroutines for Specific Periods**
- **Calling Features From Reports**



**At the conclusion of this topic, you will be able to:**

- **Call subroutines using dynamic actions**
- **Use selection reports for fast entry**
- **Call subroutines for specific periods**
- **Call features from reports**

Course Overview

Data Structures in  
Personnel Administration

The Logical Databases  
PNP/PNPCE

Join and Projection

Specific Commands

Repetitive Structures and  
List Display

Cluster Database Tables in  
HR

Data Structures in  
Personnel Planning

The Logical Database PCH

Customer-Specific Infotypes



**HR ABAP Features**

## Table of Dynamic Actions

ITy	STy	Field	FC	No	S	Variable	function part
0008	__	SPRPS	06	0	_	*-----	Lock indicator set?-----*
0008	__	SPRPS	06	1	P	P0008-SPRPS<>SPACE	
0008	__	SPRPS	06	2	F	COMPUTE_DATE(ZPFORM01)	
0008	__	SPRPS	06	3	I	INS,0019,10	
0008	__	SPRPS	06	4	W	P0019-VTRMN=RP50D-DATE1	

```
REPORT zpform01.
TABLES: RP50D,      "Return fields for dyn. actions
        PRELP.      "HR Master Data Buffer . .
* Calculate date
FORM compute_date.
    RP50D-DATE1 = PRELP-BEGDA -14.
ENDFORM.
```

© SAP AG 1999

- The 'Dynamic Actions' table (T588Z) is a control table that triggers steps taken when an infotype record is maintained. Such steps include maintaining another infotype record, performing a routine, or sending a mail.
- **OP** stands for a user operation where 02 stands for Change, 04 for Insert, and 08 for Delete a record. The values can be added, for example, 06 means that an action is performed if the current record has been changed or inserted.
- **A** stands for an action performed by the system, for example, **P** = Check a condition, **F** = Call a routine, **I** = Maintain an infotype record, **W** = Set default values when inserting a record.
- You can call internal (defined in the module pool) and external routines. In an external routine, the name of the program is given in parentheses after the name of the routine. You cannot enter the USING parameter. The fields of the structure that is not used in the standard system (**RP50D**) are available to return the values from the routine. They can be filled by the routine and then used for default values (W statements). This structure can be enhanced with customer fields in a customer include.
- In this example, the routine COMPUTE\_DATE calculates a date in the subroutine pool ZPFORM01 and places it in field RP50D-DATE1. This date is calculated by subtracting 14 days from the start date of the locked record.

```
REPORT RPLFST00 USING DATABASE PNP.
TABLES PERNR.
INFOTYPES: 0000, 0001, 0002.

DATA: BEGIN OF pernrtab OCCURS 20,
      PERNR LIKE PERNR-PERNR,
      ENAME LIKE P0001-ENAME,
END OF pernrtab.
. . .

GET PERNR.
. . .

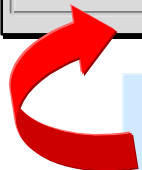
END-OF-SELECTION.
  EXPORT pernrtab TO MEMORY ID 'PERNRTAB'.
```

© SAP AG 1999

- If you use the fast entry function for master data and time data, you can use reports to effect an initial selection of personnel numbers.
- Reports that can be used for fast entry are contained in the 'Selection Reports for Fast Data Entry' table (T588R). If you intend to use customer reports for fast entry, make sure they have been entered in this table.
- The selected personnel numbers are transferred from the report to fast entry using the ABAP memory. The selection report uses the EXPORT statement to store the data as a cluster in the ABAP memory. The calling transaction then imports the data from the ABAP memory.

## Table of HR subroutines

Symb.name	Start date	End date	Module pool	Module	Type
<b>XEDT99</b>	01.01.1991	<b>31.12.9999</b>	RPCEDTX0	EXTCALL	R



```

. . .
SELECT * FROM T596F
  WHERE SNAME = 'XEDT99'
  AND   ENDDA GE PN-BEGDA
  AND   BEGDA LE PN-BEGDA.
ENDSELECT.
. . .
* Dynamic Perform
  PERFORM (T596F-MODNA) IN PROGRAM (T596F-PGMNA)
  USING . . .
    
```

© SAP AG 1999

- The HR Subroutines table (T596F) enables you to access various subroutines for specific periods in order to solve a task defined by a symbolic name.
- In the standard system, the symbolic names of subroutines all start with the appropriate country indicator.
- The symbolic name is a freely defined name (such as 'XEDT99') that is used to find a table entry. If a table entry is customer-specific, the symbolic name must start with a special character (for example, &EDT99).
- You first read the table entry valid for a particular period. This contains the name of a program in field **T596F-PGMNA** and the name of a subroutine contained in this program in field **T596F-MODNA**. You then use a dynamic action to call this subroutine from your program.

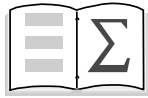


```

TABLES: PME04.           "Field string for feature ABKRS
DATA:   ret_value(2).    "Return value for feature
* Fill decision fields for feature with values.
PME04-PERSK = P0001-PERSK,
. . .
* Call feature
CALL FUNCTION 'HR_FEATURE_BACKFIELD'
  EXPORTING
    FEATURE           = 'ABKRS'
    STRUC_CONTENT     = PME04
  IMPORTING
    BACK              = ret_value
  EXCEPTIONS
    ERROR_OPERATION   = 2
    NO_BACKVALUE      = 3
    FEATURE_NOT_GENERATED = 4.
    
```

© SAP AG 1999

- This function module reads the decision tree for a feature with the accompanying field contents and determines the return values for the field contents.
- To identify which fields in the field string for the feature (PMENN) are used for the decisions in the feature, view the structure and decision tree in feature maintenance (transaction PE03).
- The field string PMENN must be declared in the data declaration part of your program using a **TABLES** statement. Define a field to contain the return value of the feature.
- In your program, enter values in the decision fields used in your feature.
- Call the function module and enter the name of the feature to be used and the name of the field string. The return value for the feature is transferred to the main program with the **BACK** parameter.
- Possible errors when processing the decision tree:
  - a) An error occurred in the feature (ERROR\_OPERATION)
  - b) No return value is available for the current contents of the decision field (NO\_BACKVALUE)
  - c) The feature was not generated (FEATURE\_NOT\_GENERATED)
- If the return value for a feature consists of a table, use the function module **HR\_FEATURE\_BACKTABLE**.



- You can use a dynamic action to call a subroutine
- You can include your own selection reports for fast entry in the system
- Subroutines can be accessed for specific periods
- Features can also be used in reports



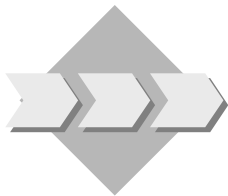
## Unit: HR ABAP Features

### Topic: Creating a Dynamic Action



At the end of this exercise, you will be able to:

- Use a dynamic action to link the maintenance of two infotypes.



If an employee from your company takes on a new function, he or she usually has a six-week introductory period. The Personnel Department would like the system to be set up so that the maintenance screen for the *Monitoring of Tasks* infotype (0019) is also displayed when the *Corporate Function* infotype (0034) is created.

- 1-1 Use the Implementation Guide (IMG) to create a dynamic action for infotype 0034 subtype 9xx (xx = group number). This action is only to be carried out when the administrator creates a new record for infotype 0034.

The new record *Monitoring of Tasks* of infotype 0019 is to be displayed with date type 06 *Introductory period* and a date. Determine the date in a subroutine pool with the formula *Start date for infotype 0034 + 42 days*.

- 1-2 Create a record for employee 50991xx for infotype 0034, subtype 9xx (xx = group number). The validity period is the current year. Test the correctness of your table entries.





**Unit: HR ABAP Features**  
**Topic: Creating a Dynamic Action**

1-1 To create a dynamic action:

*IMG → Personnel Management → Personnel Administration → Customizing Procedures  
→ Dynamic Actions → New Entries*

Make the following entries in the table:

<u>Infotype</u>	<u>Type</u>	<u>Field</u>	<u>FC</u>	<u>No.</u>	<u>A</u>	<u>Variable Function Part</u>
0034	9xx		04	0		*----- Corporate function -----*
0034	9xx		04	1	F	COMPUTE_DATE( <i>program name</i> )
0034	9xx		04	2	I	INS,0019,06
0034	9xx		04	3	W	P0019-TERMN=RP50D-DATE1

Save your entries.

Create a program with the type *Subroutine pool*.

PROGRAM *program name*.

\*-- Declaration

TABLES: RP50D, PRELP.

FORM COMPUTE\_DATE.

RP50D-DATE1 = PRELP-BEGDA + 42.

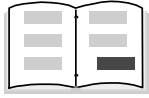
ENDFORM.

1-2 To test the dynamic action:

*Human Resources → Personnel Management → Administration → HR Master Data  
→ Display*

Choose the *Planning Data* menu and create a new infotype record.





- **This section contains additional material to be used for reference purposes.**
- **This material is not part of the standard course.**
- **Therefore, it may not be covered during the course presentation.**

## **Contents:**

- **Report categories**
- **Exporting clusters**
- **Processing payroll results in Release 4.0**
- **Data Structures in Personnel Planning: Exercises**

© SAP AG 2003



```
TABLES: PCLn.                "Import/export table
INCLUDE: RPCnxxxy0.          "Cluster definition

* Fill cluster-KEY

xy-KEY-FIELD = <VALUE>.
. . . .
* Fill data object

* Export record

EXPORT TABLE1 TO DATABASE PCLn(xy) ID xy-KEY.

IF SY-SUBRC EQ 0.
  WRITE: / 'Update successful'.
ENDIF.
```

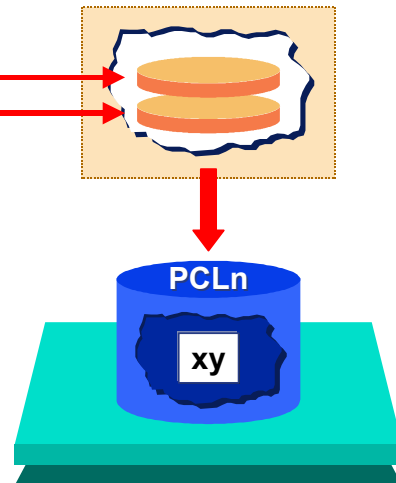
© SAP AG 1999

- The cluster definition is included using the INCLUDE statement.
- The EXPORT command writes one or more data objects with xy-KEY to cluster xy.
- If the export is successful, the return code is 0.

```

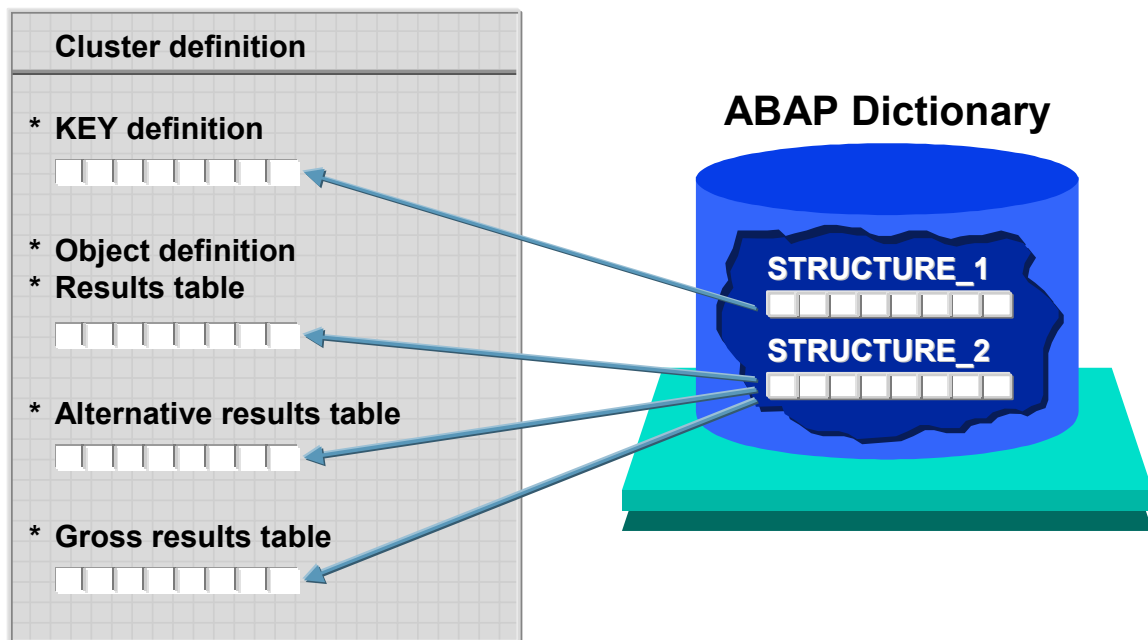
* Buffer definition
INCLUDE: RPPXD00.
DATA: BEGIN OF COMMON PART buffer.
      INCLUDE RPPXD10.
DATA: END OF COMMON PART buffer.
. . .
* Data export to buffer
RP-EXP-Cn-xy.
RP-EXP-Cn-xy.
. . .
* Save
PERFORM PREPARE_UPDATE USING 'V'.
. . .
* Buffer Administration Routines
INCLUDE: RPPXM00.
    
```

**Main memory buffer**



© SAP AG 1999

- If data is exported using macros, the data records are not written directly to the database. Instead, they are written to a main memory buffer. The data is diverted by a USING parameter in the EXPORT statement, which accesses a buffer administration routine.
- If data is exported successfully, the RP-IMP-xy-SUBRC = 0 return code is set.
- At the end of the program, the buffered records must be saved on a PCLn database.
- To save the data, the PREPARE\_UPDATE routine is accessed with USING parameter V.



© SAP AG 1999

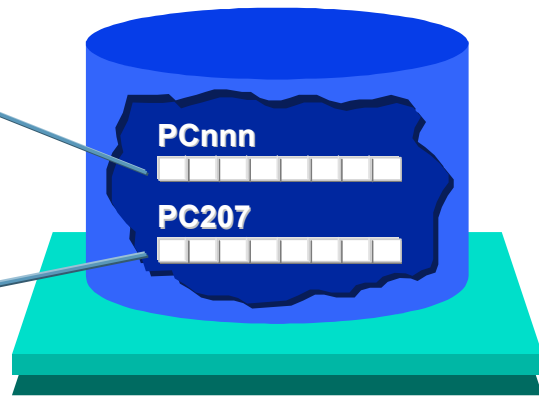
- The cluster definition of payroll results complies with the valid naming conventions. They are stored in the `RPC2xyz0 INCLUDE` reports, where `z` stands for the HR country indicator.
- The cluster key is contained in the `RX-KEY` structure.
- All of the fields in the key and data objects are defined in ABAP Dictionary structures.

```

Cluster definition
REPORT RPC2Rn00.
* KEY definition
DATA: BEGIN OF RX-KEY.
      INCLUDE STRUCTURE PCnnn.
DATA: END OF RX-KEY.

* Object definition
* Results table
DATA: BEGIN OF RT OCCURS 150
      INCLUDE STRUCTURE PC207.
DATA: END OF RT.
    
```

## ABAP Dictionary



© SAP AG 1999

- The fields for the key and objects are defined in ABAP Dictionary structures. This means that you can use them time and time again without jeopardizing consistency.
- Structures used for the cluster definition comply with the PCnnn naming convention, where nnn stands for the alphanumeric characters of your choice.
- The PC200 structure contains the key definition of the results cluster. It consists of two fields, PERNR (personnel number) and SEQNO (sequential number).
- The data definitions of international payroll results are stored in includes RPC2RX00 and RPC2RXX0.

```

DATA: number LIKE PC261-SEQNR.
. . .
CALL FUNCTION 'CD_READ_LAST'
  EXPORTING
    BEGIN_DATE      = PN-BEGDA
    END_DATE        = PN-ENDDA
  IMPORTING
    OUT_SEQNR       = number
  TABLES
    RGDIR           = RGDIR
  EXCEPTIONS
    NO_RECORD_FOUND = 1
    OTHERS          = 2.
. . .
RX-KEY-PERNR = PERNR-PERNR.
RX-KEY-SEQNR = number.
RP-IMP-C2-RX.
. . .
LOOP AT RT.
  WRITE: / RT-LGART...
ENDLOOP.

```

© SAP AG 1999

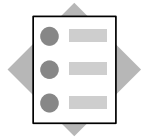
- This function module determines the current payroll result for a for-period to be evaluated. To determine the correct start date and end date of the for-period, you specify the period by entering the payroll period in the selection screen. If you specify report class XXM00004 in the attributes of your report, the payroll period is entered and the start date (PN-BEGDA) and the end date (PN-ENDDA) are determined using the Payroll Periods table (T549Q).
- You enter the start and end date of the for-period for the evaluation as well as table RGDIR. The function module then gives you the sequential number (OUT\_SEQNR) for the current (A) result of the for-period.
- You then transfer the sequential number to RX-KEY and import the required payroll result with the corresponding macro RP-IMP-C2-xy (xy = cluster ID).
- You process the wage type tables for the payroll result sequentially using LOOP.
- You can also use the following function modules:
  - CD\_READ\_PREVIOUS  
(reads the record that precedes the payroll record)
  - CD\_READ\_PREVIOUS\_ORIGINAL  
(reads the last original result that precedes the original payroll result)

```

*Table containing evaluation periods
DATA: BEGIN OF EVPDIR OCCURS 100.
      INCLUDE STRUCTURE PC261.
DATA: END OF EVPDIR.
...
CALL FUNCTION 'CD_EVALUATION_PERIODS'
  EXPORTING
    BONUS_DATE      = <Bonus date>
    INPER_MODIF     = <Period par.>
    INPER           = <In-period>
    PAY_TYPE        = <Payroll type>
    PAY_IDENT       = <Payroll ID>
  TABLES
    RGDIR           = RGDIR
    EVPDIR          = EVPDIR
  EXCEPTIONS
    NO_RECORD_FOUND = 1
    OTHERS          = 2.
    
```

© SAP AG 1999

- This function module fills table EVPDIR (evaluation periods) with payroll result data from the selected period. Descriptive data from the current (A) and previous (P) result is written to table EVPDIR.
- **Period parameter:** Specifies the period in which the payroll runs for a payroll area. Just one period parameter is assigned to a payroll area. Payroll periods are defined for each period parameter.
- **In-period:** The period in which a payroll result is created.
- Payroll type: Indicator for "Type of Payroll Result" (according to table T52BX)
  - " " = Regular payroll result (payroll result for a payroll period)
  - "A" = Bonus accounting (bonus date transferred)
  - "B" = Correction run
  - "C" = Manual check
- **Payroll identifier:** Indicator used to make a distinction between various special accounting runs performed on the same day
- You can also use the following function modules:
  - CD\_READ\_PREVIOUS  
(reads the record that precedes the payroll record)
  - CD\_READ\_PREVIOUS\_ORIGINAL  
(reads the last original result that precedes the original payroll result)



**At the conclusion of this topic, you will be able to:**

- **Describe the special features of external object types**
- **Describe the special features of external infotypes**

- **Definition of external object types and relationships**

- **External object types** = Object types that are not stored in Personnel Planning structures (that is, no record in HRP1000), but are only referred to in the context of relationship records (only records in HRP1001)

- **Table T77EO** (IMG path: *Maintain Object Types - External Object Types*)

Object type	Key structure	Inversion	Interface
P	PKEYS	X	RHPREL00
K	PKEYK		RHKOST00
...			

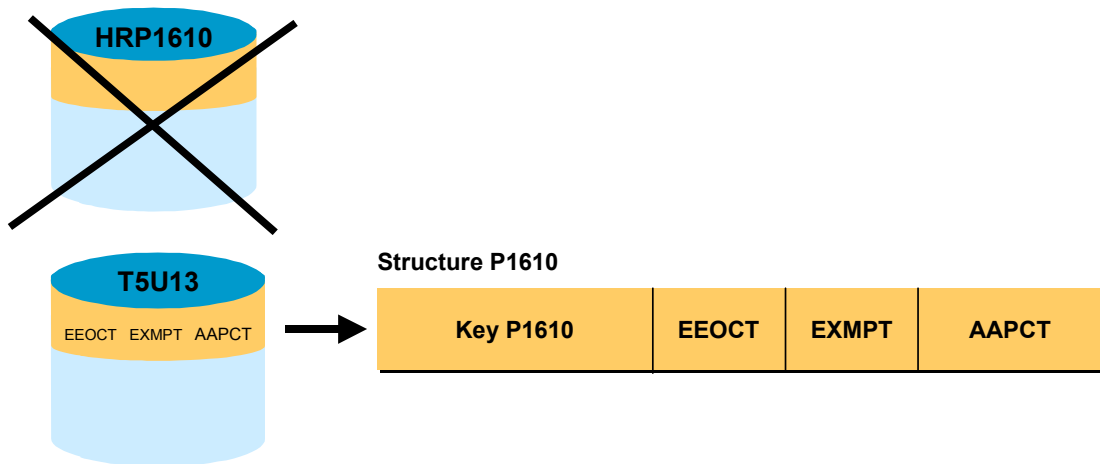
Depending on the key structure, a distinction is made between external object types with a physical inverse relationship in HRP1001 (for example, persons from PA) or without (for example, cost centers from CO).

- **External relationships** = Simulation of complete relationships, that is, no relationship direction is physically stored in HRP1001 (**table T77EV**; IMG path: *Maintain Relationships - Allowed Relationships - External Relationships*)



- **Definition of external infotypes**

- ➔ **External infotypes** = Simulation of information on Personnel Planning objects that are only stored in Administration Customizing tables and not as a Personnel Planning infotype in transparent infotype tables HRPnnnn
- ➔ For example: Additional job data for the USA, infotype 1610 and T5U13

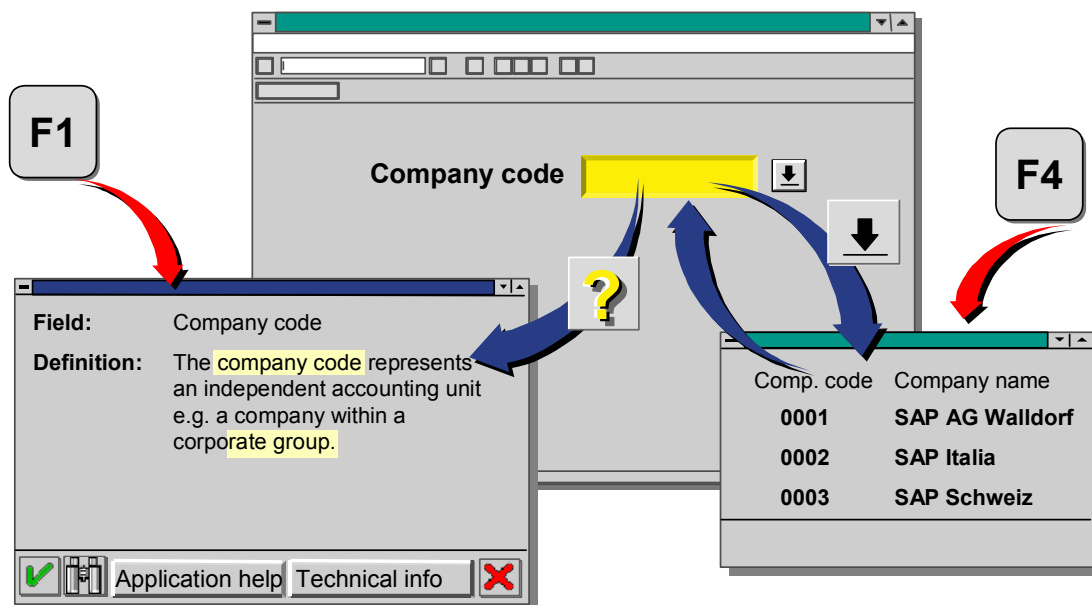


© SAP AG 2003



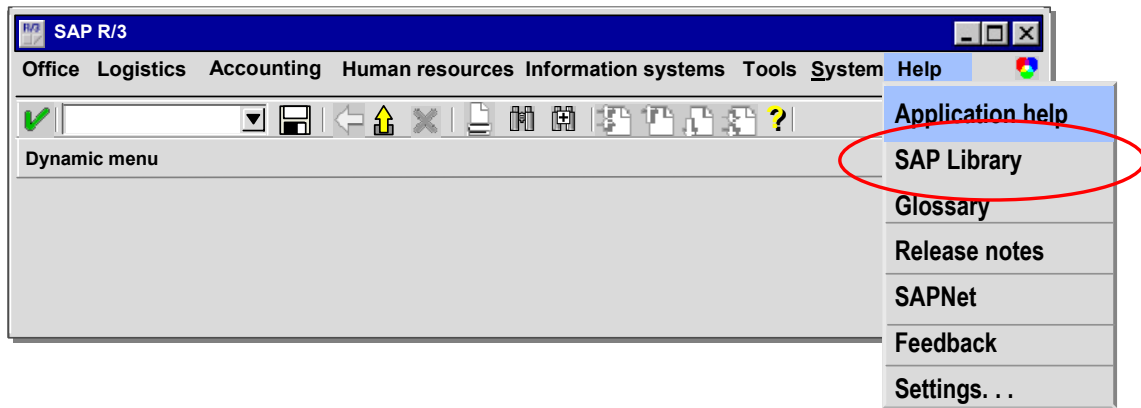
## **Contents:**

- **Human Resource Process**
- **Enterprise Structure**
- **Personnel Structure**
- **Infotypes**
- **Time Management**
- **Payroll**



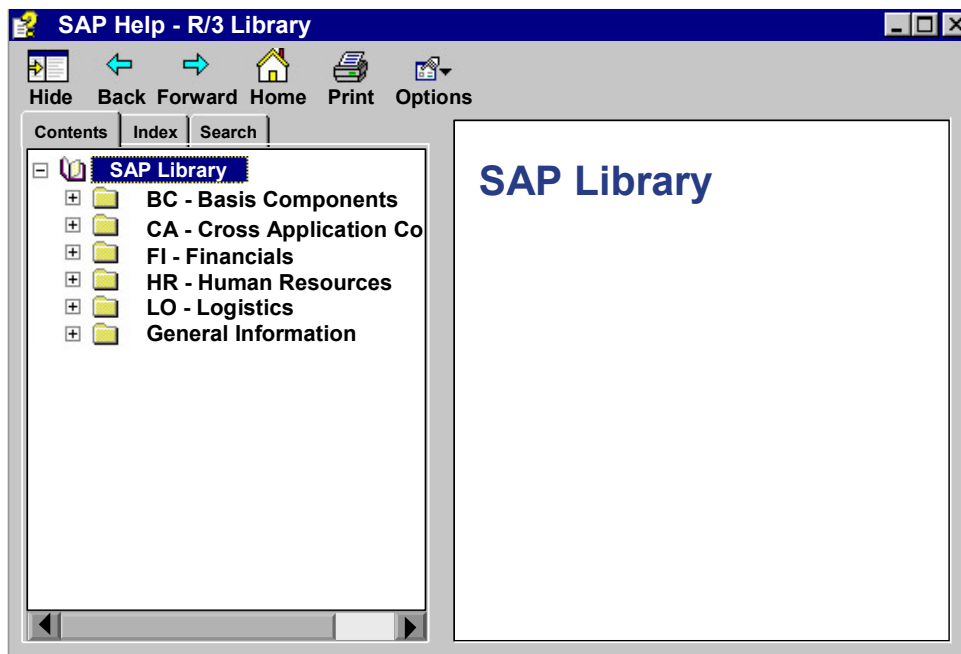
© SAP AG 1999

- For help on a field, place the cursor on the field and press [F1] or choose the help symbol - a question mark.
- You can access glossary information for terms highlighted in the help text.
- To do this, point to the term and click on the highlighted term.
- For further information on possible entries and the field definition, choose *Application Help* or *Technical info*.
- Access a list of possible entry values or search helps to help you search for the data you require, by placing the cursor on the entry field and pressing [F4] or choosing the possible values icon at the end of the field.
- To transfer a value from the list of entry values to the entry field, place the cursor on the value and choose *Choose*, choose [F2], or point and double-click the left mouse button.



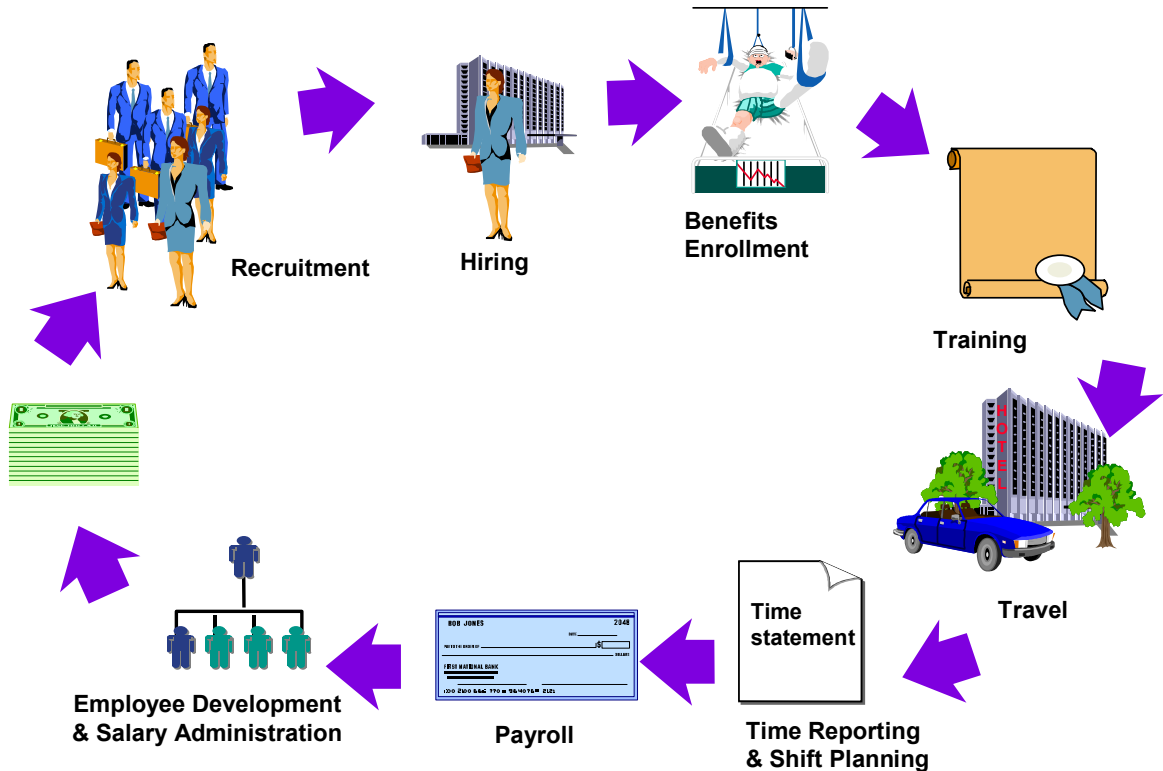
© SAP AG 1999

- There are many ways to access help within the SAP System.
- *Getting Started* is a good introduction to the basic concepts of the SAP system, for example, logging on and off, using the SAP System, working with several sessions, and so on.
  - It describes common screens.
  - It describes common actions.
- *Getting Started* is part of the SAP Library.
  - It is in HTML format.
  - It contains links to other help sections.
- *Release Notes* explain what has changed between releases of the SAP System.



© SAP AG 1999

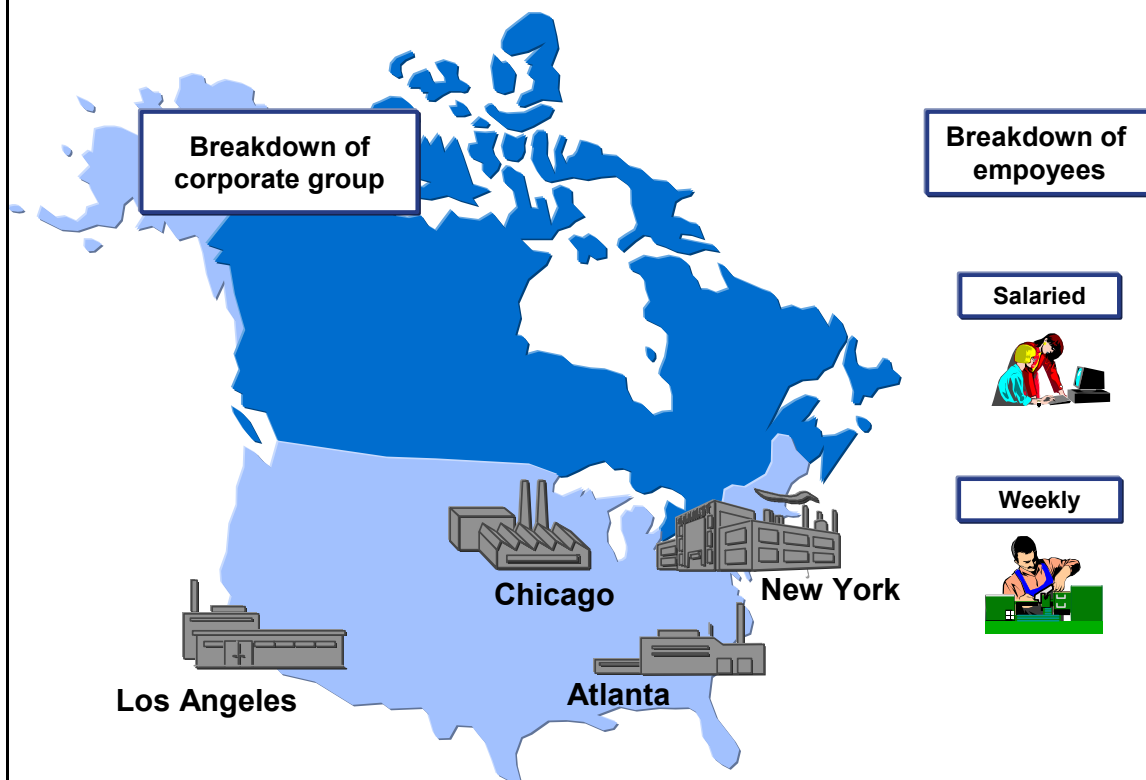
- Most SAP Help files are HTML files.
- *SAP Library* is an online library of the entire R/3 documentation. You can search for the information you need or drill down through applications.
- *Application Help* displays specific information from the SAP Library. The information displayed is determined by which screen you are in.
- The *Glossary* defines technical terms used within the SAP system.



© SAP AG 1999

■ Every component of this process is supported by the SAP Human Resources system:

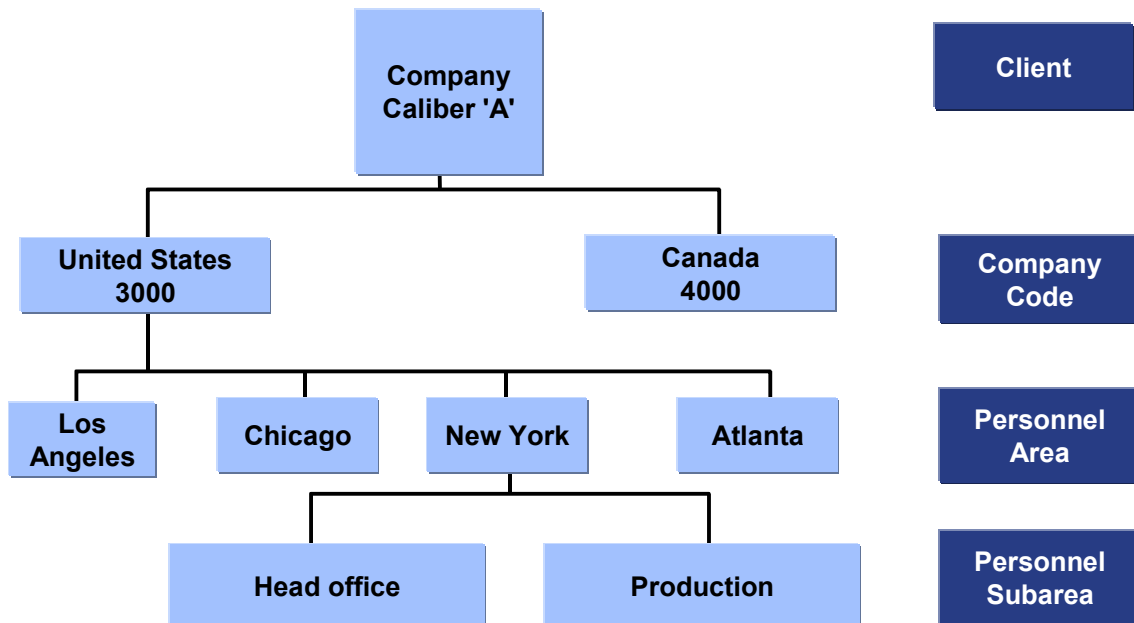
- Recruitment
- Hiring
- Benefits Enrollment
- Training
- Travel Expenses
- Time Evaluation
- Shift Planning
- Payroll
- Employee Development
- Compensation Management
- Cost Planning
- Reporting



© SAP AG 1999

- A personnel management system should enable you to reproduce company hierarchies and the relationships within the hierarchies, and to store and manage employee data.
- The organizational hierarchy refers to the division of a corporate group into company code, personnel areas, and personnel subareas.
- The first step in recording personnel data in the SAP System is to assign the employee within the company structure.
- The employee is assigned to a work area, a specific status (for example, active or inactive) and type (for example, salary or hourly).
- Employees are also assigned a position within the organization - within employee groups and employee subgroups.



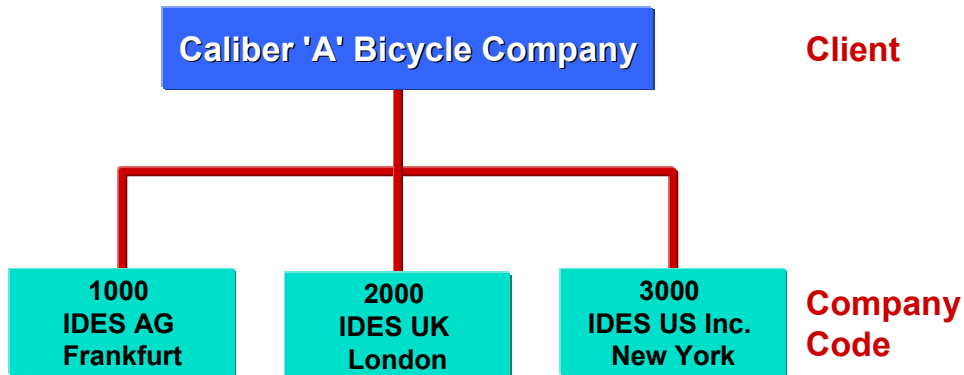


© SAP AG 2003

- The company structure for Human Resources consists of the following:
  - Client
  - Company code
  - Personnel area
  - Personnel subareas
- A **client** can be valid for one or more company codes.
- A **company code** is defined in Accounting. Legally required financial statements, such as balance sheets and profit and loss statements, are created at the company level.
- The **personnel area** is used exclusively in Human Resources and is unique within a client. Each personnel area must be assigned to a company code.
- The final element of the company structure, unique to Human Resources, is the **personnel subarea**.

**The client is an independent legal and organizational unit of the system.**

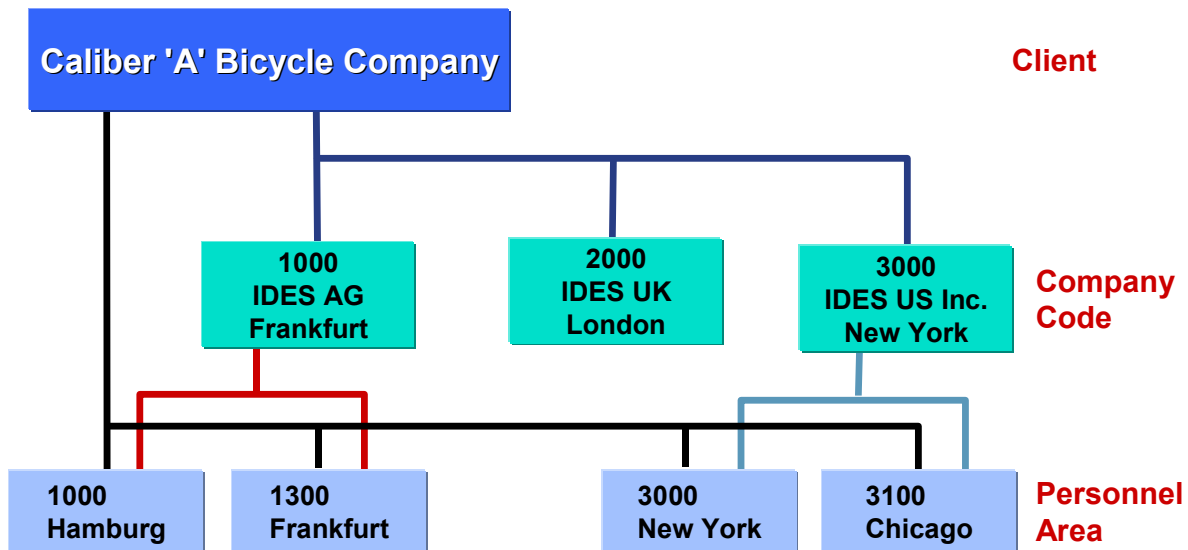
**The company code is an independent company with its own accounting unit - a company which draws up its own balance sheets.**



© SAP AG 1999

- In legal and organizational terms, the client is a self-contained unit within the system. You should consider the following points before you decide whether or not to set up a client:
  - It is not possible to access personnel data in other clients; this protects data from being accessed by unauthorized users.
  - You cannot assign access authorization or run evaluations for all clients.
  - There is no exchange of data between clients.
  - If an employee changes clients, you have to create the personnel number again.
- The company code is a self-contained unit in legal terms, for which you can draw up a complete set of accounts. The company code is the highest level of the company structure. If you also use the Controlling, Financial Accounting, Materials Management or Sales and Distribution components, the company code must be set up in consultation with these applications. This does not impose any restrictions, however, as all important control information for the Human Resources system is defined at personnel subarea level.
- All of the Caliber 'A' Bicycle Company's United States employees belong to company code 3000.

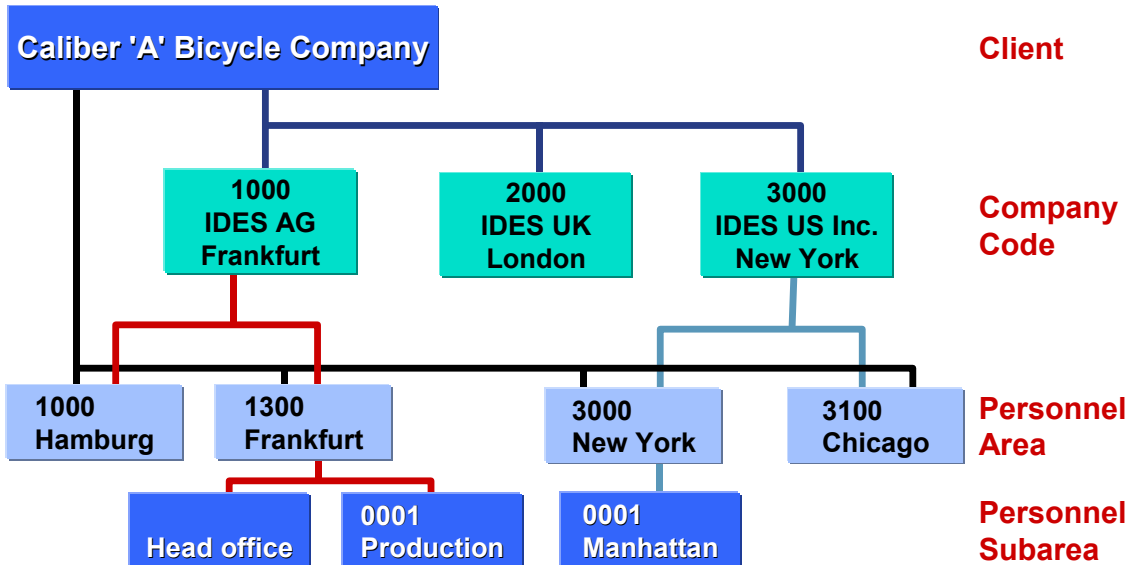
A personnel area is a Personnel Administration-specific unit and is the subunit of the company code.



© SAP AG 1999

- A **personnel area** is a Personnel Administration-specific unit and is the subunit of the company code. The individual personnel areas in a company code have four-digit alphanumeric identifiers.
- The personnel area has the following functions:
  - It allows you to generate default values for data entry, for example, for the payroll area.
  - It is a selection criterion for reporting.
  - It constitutes a unit in authorization checks.

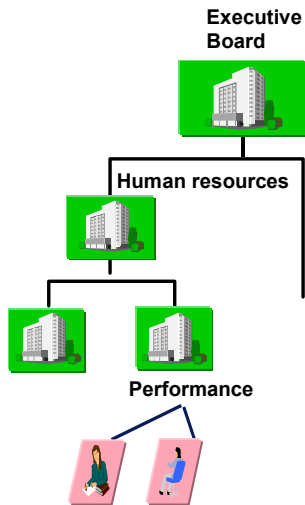
Personnel subareas are subdivisions of personnel areas.  
The organization of the most important subareas of personnel administration takes place at this level.



© SAP AG 1999

- **Personnel subareas** are subdivisions of personnel areas. The organization of the most important subareas of personnel administration takes place at this level. The personnel subarea is assigned a four-character alphanumeric identifier. The control features are stored according to the country.
- The main organizational functions of the personnel subarea are as follows:
  - To specify the country grouping. Master data entry and the setting up and processing of wage types and pay scale groups in payroll depend on the country grouping. The grouping must be unique within a company code.
  - To assign a legal person which differentiates between companies in legal terms.
  - To set groupings for Time Management so that work schedules and substitution, absence and leave types can be set up for individual personnel subareas.
  - To generate a default pay scale type and area for an employees basic pay.
  - To define a public holiday calendar.
  - To define subarea-specific wage types for each personnel area.

## Organizational view



## Administrative view

Organizational Assignment (0001)

PersNo 1000  
Name Anja Muller  
Financial accounting

01.05.1998 - 31.12.9999

Enterprise structure

Pers. structure

Org. structure

Position

Percentage 100

Job 50000064

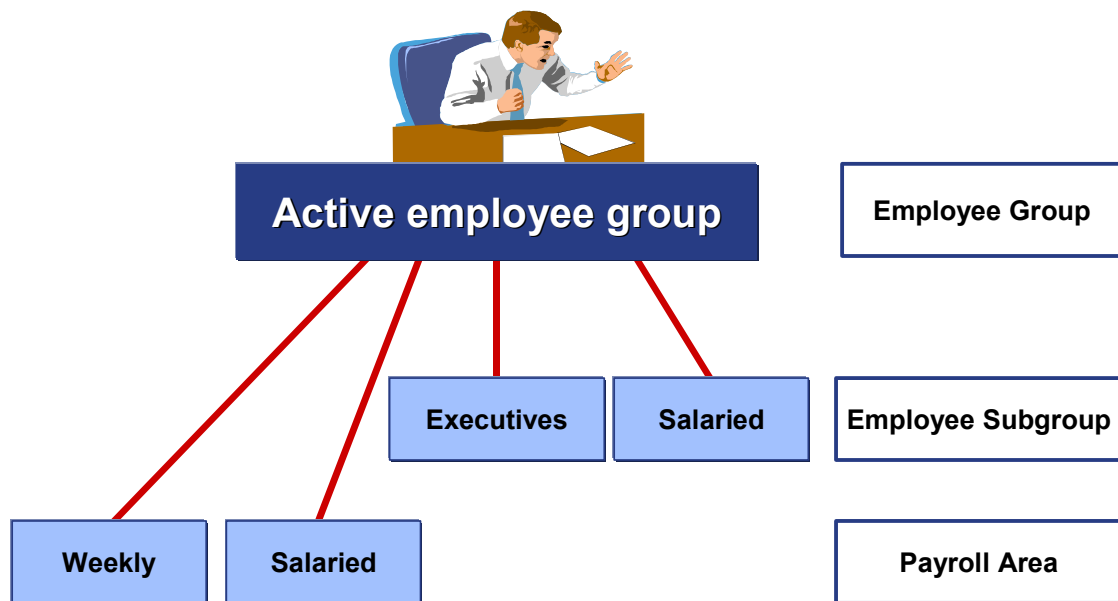
Org.unit 50000827

Org.key 0001

Administrator

© SAP AG 1999

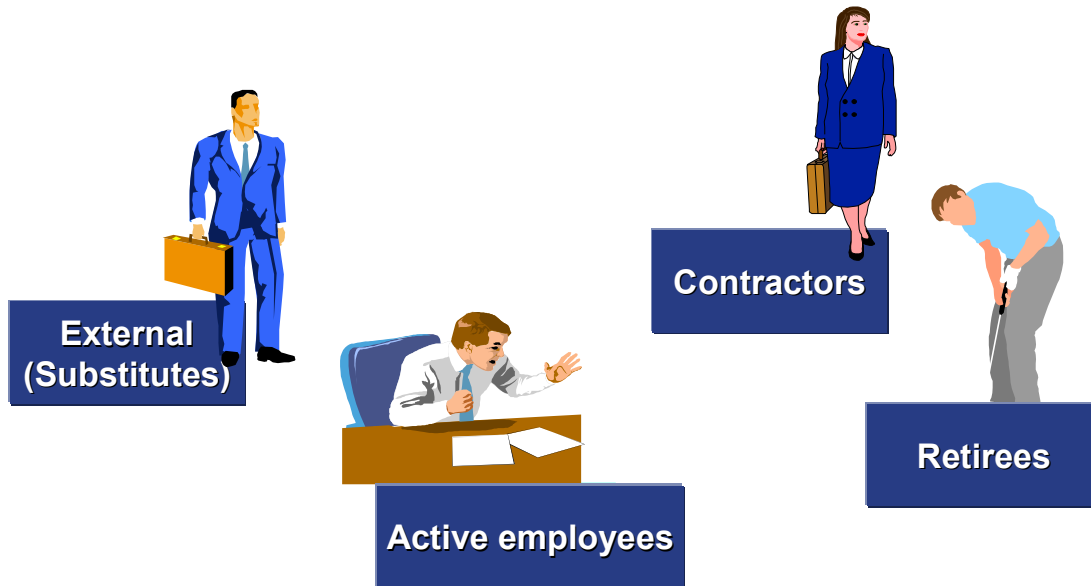
- When an employee is hired, he or she is assigned to an organizational unit, a job and a position in the *Organizational Assignment* infotype (0001).
- Organizational unit, job, and position are all components of the organizational plan and are maintained in the *Organizational Management* component.
- You implement the *Organizational Management* component to create a model of your enterprise in the system.



© SAP AG 1999

- The administrative personnel structure is made up of three elements:
  - Employee groups
  - Employee subgroups
  - Payroll areas
- Part of the personnel structure for the Caliber 'A' Bicycle Company is shown above.
  - The active employees belong to the employee group Active.
  - The employee group Active Employees is subdivided into different employee subgroups, for example, salaried employees, trainees, and monthly-wage earners.
  - A payroll area groups together employees for whom payroll is to be run at the same time.
- In the new Customer Service Center of the Caliber 'A' Bicycle Company, all employees will be active. Managers and supervisors will be salaried and customer service representatives will be hourly employees. All employees will receive their checks on a semi-monthly basis.

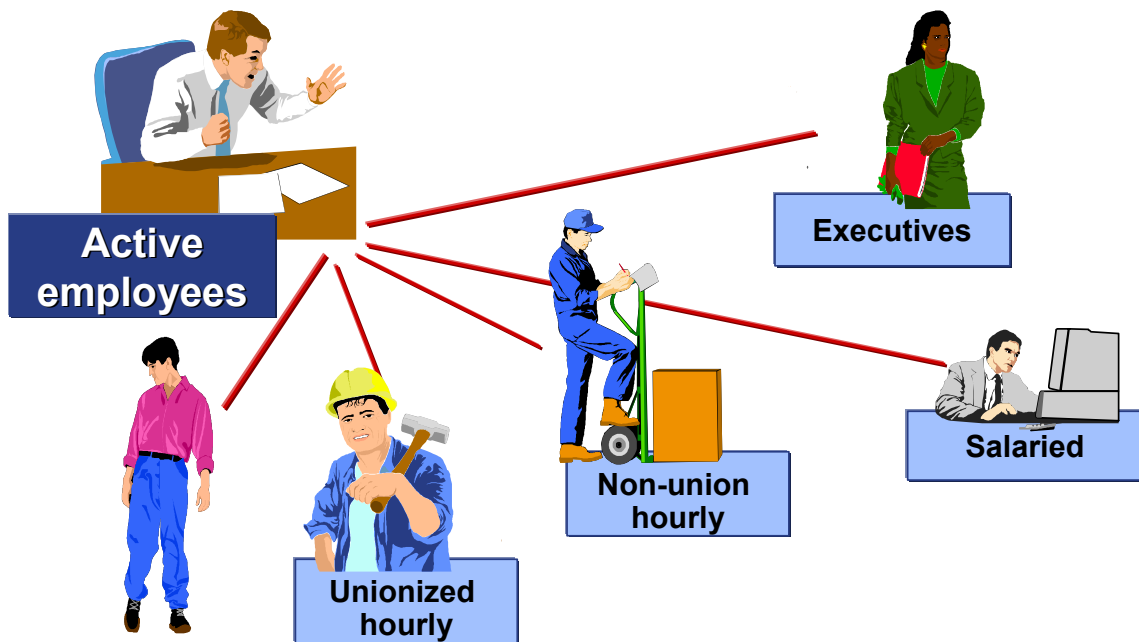
Employees are categorized into various employee groups.



© SAP AG 1999

- Employee groups represent different types of employees in an organization.
- The Caliber 'A' Bicycle Company has active employees, retirees, contractors, and external employees.
- Employee groups are used to:
  - Generate default values for payroll area and basic pay
  - As a selection criterion for reports
  - As one unit of the authorization check
- You can use the system-delivered entries to set up employee groups and extend this list to suit particular organizational requirements.

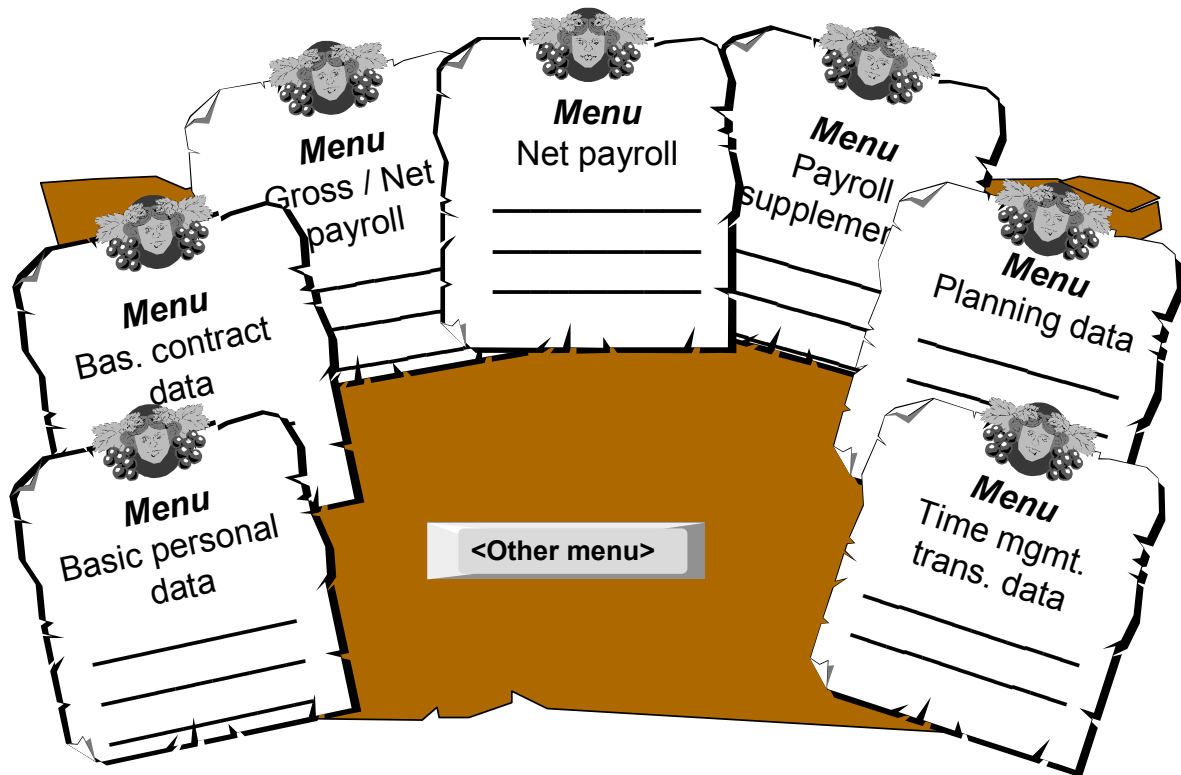
## Employee groups and employee subgroups



© SAP AG 1999

- The employee subgroup is a division of employee groups according to the types of employees.
- The employee subgroup determines:
  - Payroll calculation
  - Work schedule groupings
  - Grouping for personnel calculation rule (Payroll)





© SAP AG 1999

- Infotypes are used in HR to input and store related employee information.
- The most frequently used infotypes are grouped according to function and are contained in menus. One infotype can occur in several menus. Infotypes that are seldom used are not contained in menus, however they can be accessed individually by entering the four-digit infotype number.
- Select the required infotype menu directly using the tab page. If the required menu is not displayed, you can use the forward or backward arrow to search or you can use the pulldown menu to find the required menu.



**Maintain Master Data**

**Personnel number** 1000  
Anja Muller

**Basic personnel data**

☐ Actions  
☐ Personal Data  
☐ Addresses

**Period**

**Direct selection**

**Infotype**  **Type**

 An illustration of a right hand with an orange skin tone, pointing towards the "Period" field of the SAP screen. The hand is wearing a light blue sleeve.

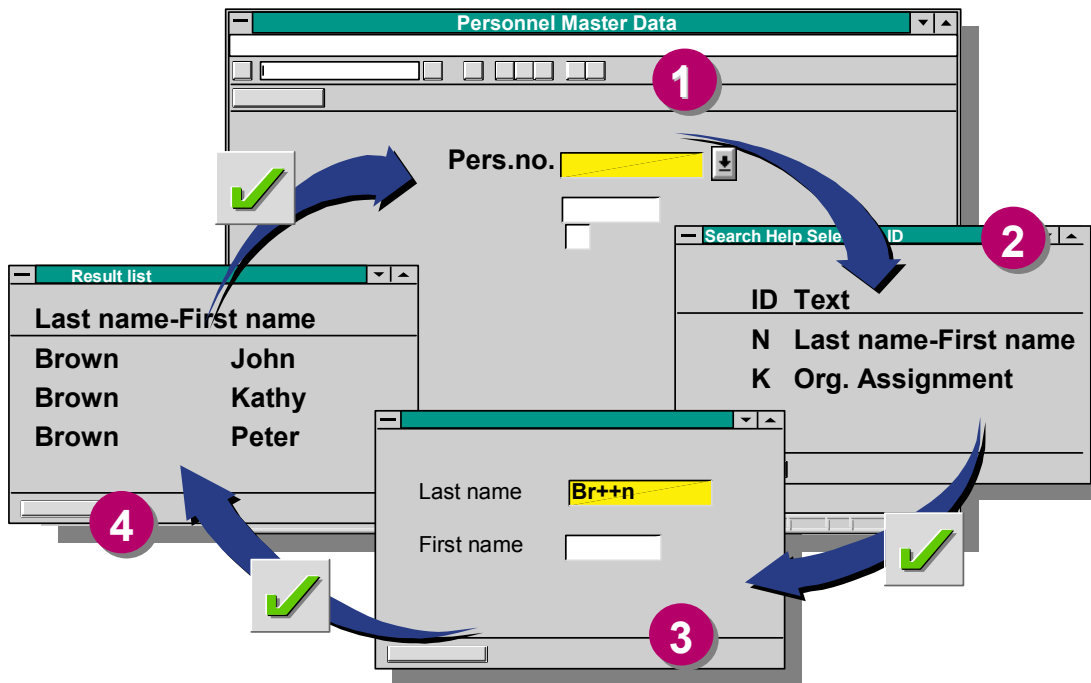
© SAP AG 1999

■ To access a selection screen with an infotype menu:

1. Choose the menu path:

Human resources → Personnel management → Administration → Display master data

2. Enter a personnel number to display information about the specified employee.



© SAP AG 1999

- There are a number of search utilities consisting of various search strings arranged in a particular order. Each search utility has its own identification (ID).
- To search for a personnel number using a matchcode, place the cursor on the personnel number entry field and click on the Possible entries button.
- A list is displayed showing all the available search utilities. You can then choose the required matchcode by double-clicking on it.
- In the subsequent screen you enter search terms, such as last and first name.
- You will then obtain a list of all personnel numbers that meet the search criteria. You can choose the personnel number you require from this list.

**Basic personnel data**

☐ Events

☐ Org. Assignment

☐ ...

☐

**Infotype**

6

**Display**

**Display Addresses (Infotype 0006)**

Pers.no.

Valid

Address type

**Address**

c/o

Street and house no.

City, state and zip

© SAP AG 1999

- If you know the number of the infotype, you can enter it in the Infotype field. Choose the requested processing type and the system will display the selected infotype record.



## **Change (Correct) → No History**

- Correct errors
- Complete missing fields



## **Create → History**

- Create a new record
- Empty input mask
- Fill in entry fields



## **Copy → History**

- Create a new record
- Complete mask from the previous record
- Customize fields
- Specify start date

© SAP AG 1999

- There are three functions to maintain infotypes: change, create, and copy.
- By using the **Change** function, you can correct an existing record without creating a new one. You choose this function if you want to correct the field contents or add additional data. This function does not create a new record, thus no history is available.
- If an employee's personal data has changed, choose the **Create** function to enter new data. A blank input template is displayed. If you create a new record, the old record remains in the system and is delimited if necessary. An infotype history is generated according to the validity periods for the individual records.
- You can also use the **Copy** function to create a new record. The system displays the current record together with the field contents. You can either copy the field contents that has not changed or, if necessary, maintain the data; this means that you do not have to enter the data in a blank template. To continue the infotype history, enter the start date of the new record.
- When you make changes relating to personnel numbers stored in the system, you should always create new records to keep the infotype history up to date. For example, you create a new record when an employee's address changes: the old address is valid until December 31, 1998 - the new address is valid from January 1, 1999.



### **Save**

You must save each infotype record you change or create.



### **Delete**

Allows you to delete entire infotype records.



### **Previous record / Next record**

Allows you to scroll between records within an infotype.



### **List**

Allows you to display a list of all records for an infotype in chronological order.

### **Maintain text**

For information purposes, you can store a text for each infotype record.

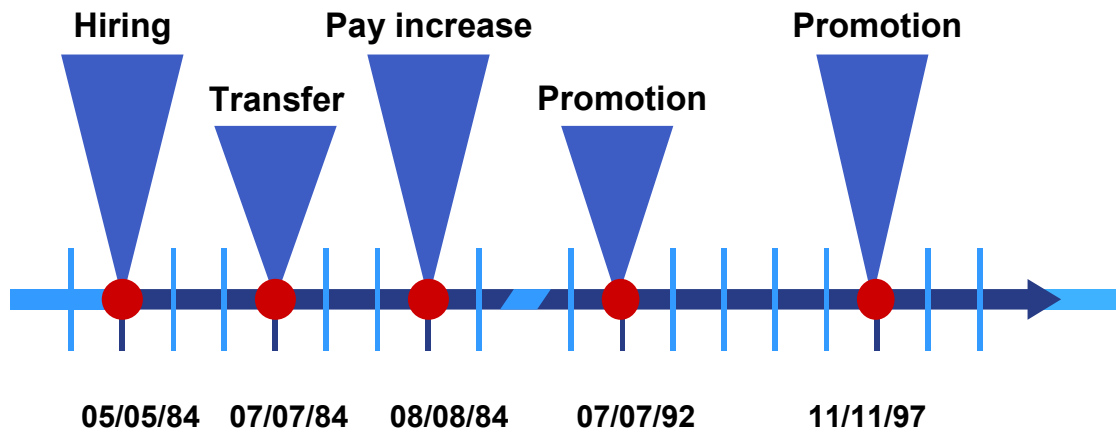


### **Overview**

Also displays infotype records in chronological order.

© SAP AG 1999

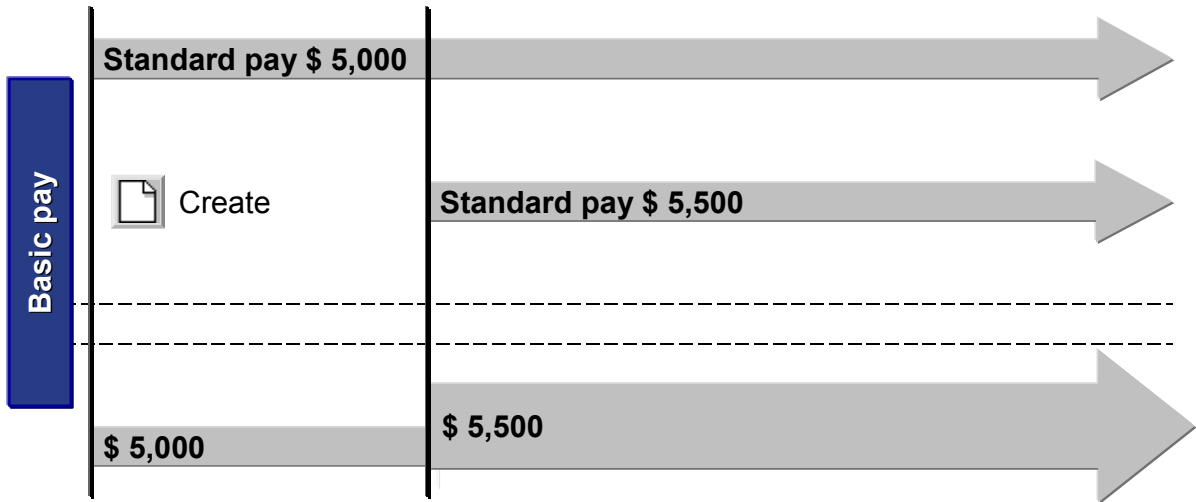
- Use the icons when editing infotypes.



© SAP AG 1999

- When you create an infotype, the old data is not lost, but is stored as history for reporting purposes. Entering a specific validity period for each infotype record allows several records to exist at one time for an infotype.

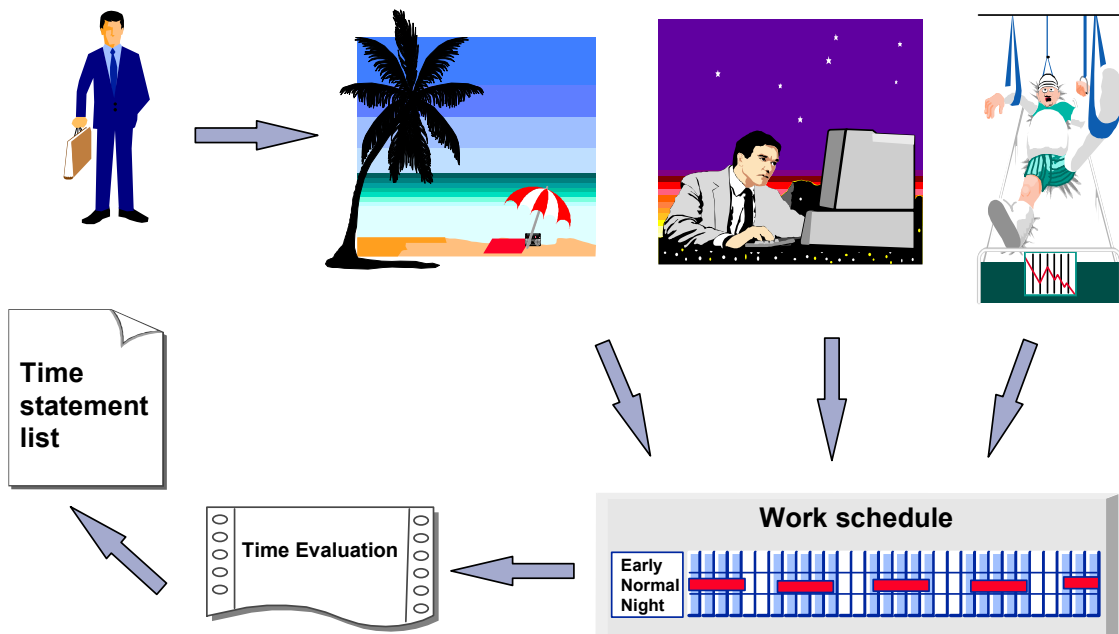
01/01/1996                      07/01/1996                      12/31/9999



© SAP AG 1999

- When you create a new record that must cover the time axes without gaps and overlaps (time constraint 1), the existing record uses the day before the start date of the new record as the validity end date. The new record is then valid until the year 9999.
- If you delete a record that must exist in the system at all times, the previous record is automatically extended.
- You can make a retroactive payroll change and all corrections are applied to the system.





© SAP AG 1999

- Employees are assigned to a work schedule, which specifies the times at which they should be at work.
- There can be exceptions to the work schedule due to vacation or sickness, for example. Exceptions must also be recorded in the system.
- Each employee's working times are evaluated, and he/she is paid accordingly.
- A time statement list can be printed for each employee at the end of the month.

# Time Management

Time recording

Time evaluation

Transfer employee time data to the SAP system

Valuate and account time data



Time  
recording



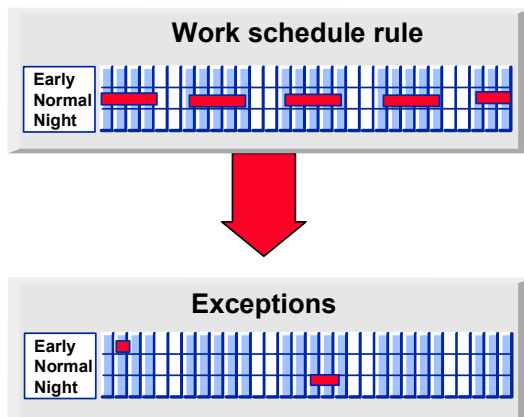
HR report

© SAP AG 1999

- HR Time Management deals with the recording and evaluation of employee time data.
- Times are recorded with a time recording terminal. The recorded times are transferred to Time Management without further valuation.
- The time data is evaluated and remunerated using a report.

## Negative recording

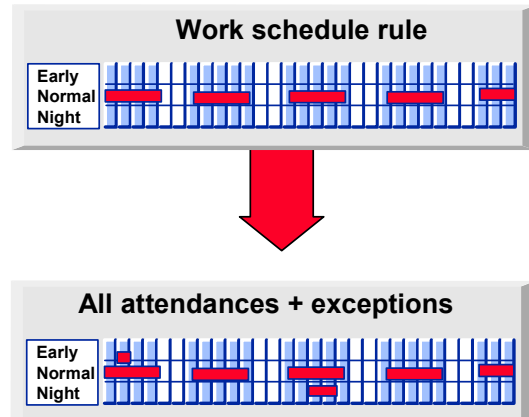
**Record exceptions to the work schedule**



## Positive recording

**Record actual time**

- Automatically
- Manually



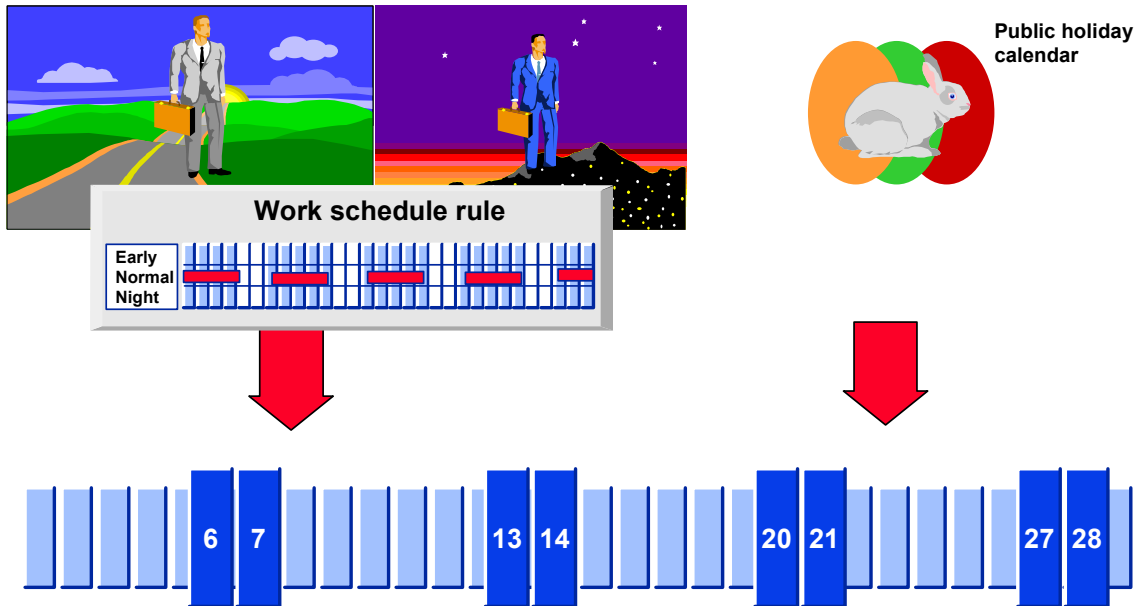
© SAP AG 1999

### ■ There are two methods for recording times:

- Recording exceptions to the work schedule: only time data that is different from the specifications in the employee's work schedule is recorded; for example, sickness, substitution, and vacation.
- Additional recording of actual times: all the employee's attendance times (actual times) are recorded, as well as any exceptions to the work schedule.

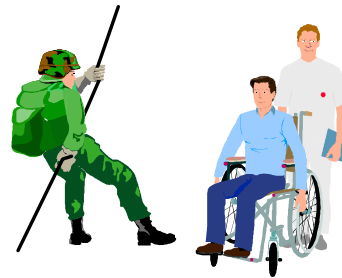
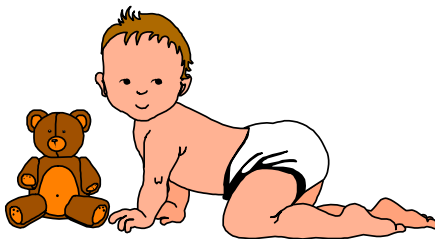
### ■ Actual times can be recorded in two ways:

- Automatically:  
The actual times are recorded in the time recording terminal and are then transferred to the SAP system. They are then processed by a time evaluation program.
- Manually:  
Actual times can be manually entered in the Attendance infotype (2002).



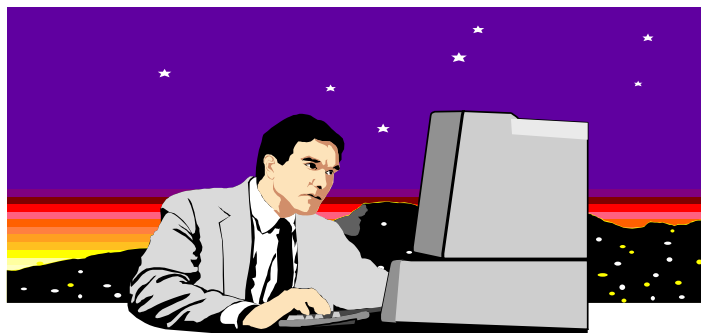
© SAP AG 1999

- To work with the Time Management system, two items are required:
  - A calendar of public holidays
  - A work schedule, which specifies daily and monthly working patterns



© SAP AG 1999

- You can record employee absences in infotype 2001. Employees are considered absent if they do not complete the planned working time specified in their work schedule.
- Absences are grouped into various absence types, which form the subtypes of the Absences infotype (2001):
  - Leave
  - Sickness
  - Family medical leave
  - Military/non-military service, and so on.
- There are a variety of different entry screens in infotype 2001, depending on whether you are recording a general absence, an absence with quota deduction, or a period of sickness or leave.

**Accounting/Logistics data**

- Activity allocation
- Cost assignment
- External services

**Alternative payment**

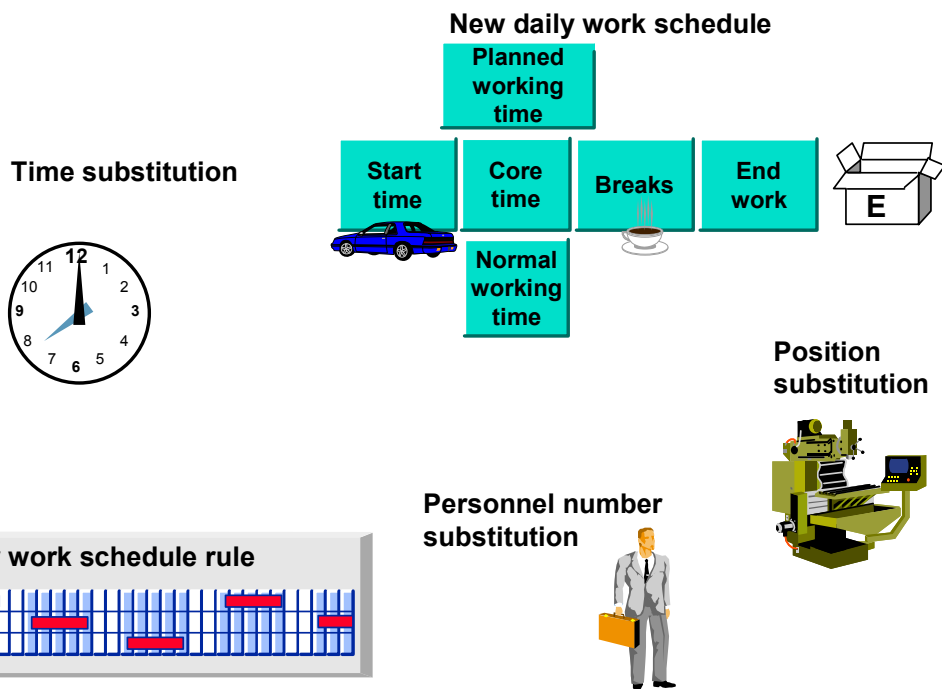
Premium  
Pay scale group/level  
Extra pay ....

**Cost assignment**

Cost center  
Order  
....

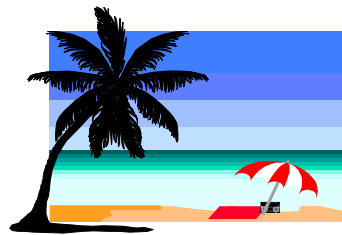
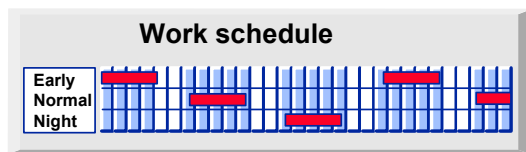
© SAP AG 1999

- Overtime is time worked in addition to the planned working time defined in the employee's work schedule rule:
  - Recorded in the Overtime infotype (2005)
  - Calculated automatically if you work with time evaluation
  - Always outside of the employee's planned working time
- Records can be for a period of one or more days.



© SAP AG 1999

- Changes to regular working hours are recorded in the Substitutions infotype (2003).
- Standard substitution types include:
  - Substitution with the individual daily work schedule
  - Substitution with the daily work schedule
  - Substitution with the work schedule rule
  - Substitution with the work schedule rule assigned to another employee (personnel number substitution)
  - Substitution with the work schedule rule assigned to another position (position substitution)
- Substitutions represent different working times for an employee. Position substitutions involve a different rate of payment, but the same working time.



Exceptions to  
work schedule



Personal  
work schedule

Overview of daily  
work schedules

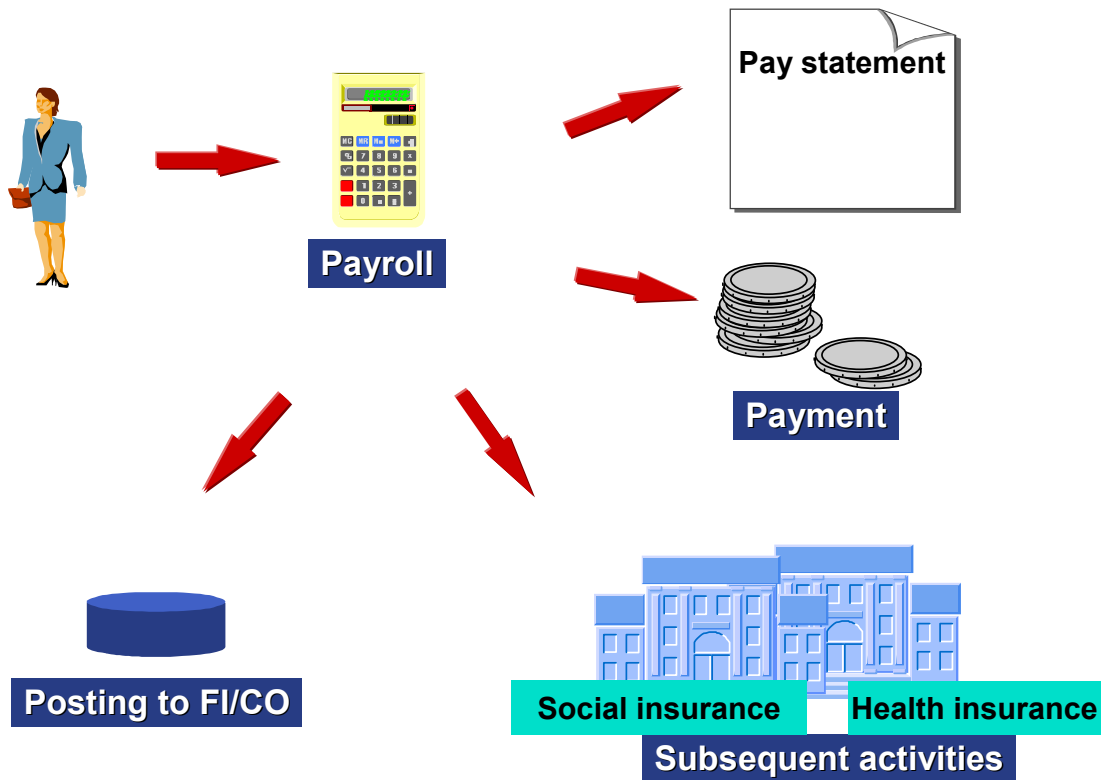
Absences  
per employee

Absences per  
organizational  
unit

© SAP AG 1999

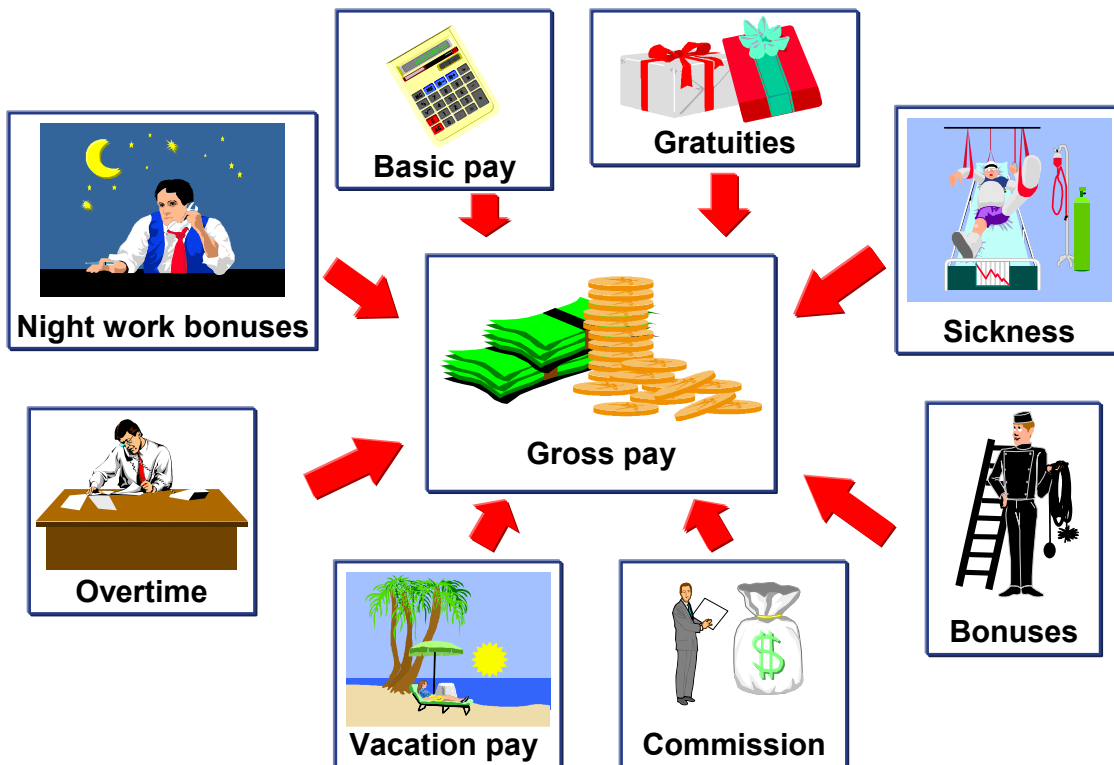
- You can run various reports in Time Management based on monthly work schedule data and attendance/absence data.





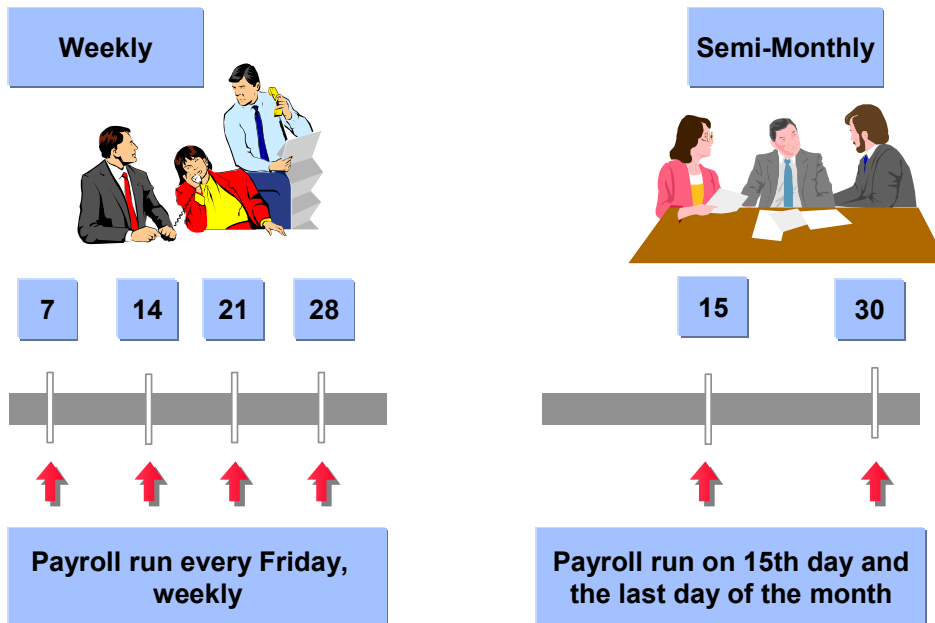
© SAP AG 1999

- An employee's pay data is evaluated in a payroll run. The result is that an employee receives a remuneration statement and the amount is transferred or paid by check.
- After payroll has been run and verified, subsequent activities can be carried out. This includes direct deposit transmissions, tax filings, and posting results to accounting.



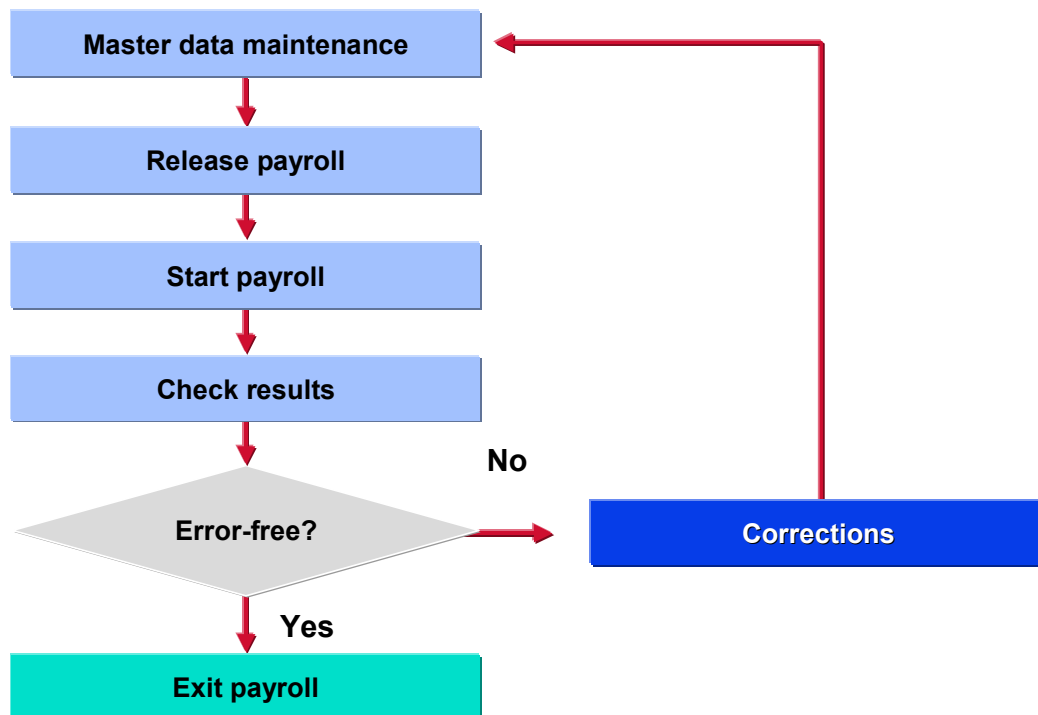
- The amount of an employee's gross pay is calculated on the basis of individual wage types used in a payroll period.
- Payments that may be included in the calculation of gross pay are basic pay, bonuses, vacation pay, Christmas bonuses, gratuities, and so on.
- Deductions might include United Way, a medical or bond deduction, for example. A wage type is a four-digit code that represents an earning or a deduction. Wage types can be entered on-line or generated during the payroll run using rule tables.

**A payroll area is an organizational unit for payroll**



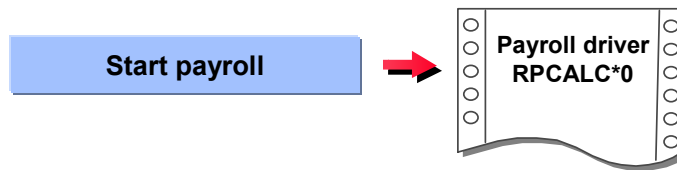
© SAP AG 1999

- The payroll program is usually run separately for different employee groups. A typical example is that of salaried employees and hourly employees, for whom payroll is performed using two different payroll periods. The payroll area determines when payroll is run.
- The payroll area serves two functions, both of which are prerequisites for running the payroll. The payroll area groups together personnel numbers that must be processed in the same payroll run
- The payroll area determines the payroll period.
- Employees for whom payroll is performed at the same time and for the same payroll period are assigned to the same payroll area. Personnel numbers are assigned to a payroll area in the Organizational Assignment infotype (0001).



© SAP AG 1999

- The payroll control record keeps track of each payroll run to ensure that each is processed sequentially.
- When the master data has been maintained, the next step is to release payroll.
- When payroll has been released for a payroll area, you cannot make changes to past or present data that would affect personnel numbers assigned to the respective payroll area. You can still make changes that apply to a future date.
- When a payroll run is started, you are not permitted to make any changes to master or time data that affect the payroll, past or present. The payroll control record ensures that payroll runs and maintenance activities do not conflict.
- If you set the status to Check result, it is not possible to make any changes that affect the payroll, past or present. It is also not possible to make changes to Customizing tables or to the start data for the payroll run. The payroll run is frozen until you release the run for payroll or for corrections.
- The correction status releases the employee's data. You can then enter changes for the payroll period and for the previous periods. If data in the payroll past has been changed, the system automatically triggers a retroactive run for the employees in question.
- You cannot activate this particular parameter until payroll is complete for all personnel numbers in the selected payroll area.



The screenshot shows the "Payroll Driver" selection screen. At the top, there is a title bar "Payroll Driver" with a dropdown arrow. Below it is a toolbar with icons for selection, help, and other functions. The main area is divided into two sections: "Payroll period" and "Selection".

**Payroll period**

Payroll area: U1

☒ Current period

☐ Other period

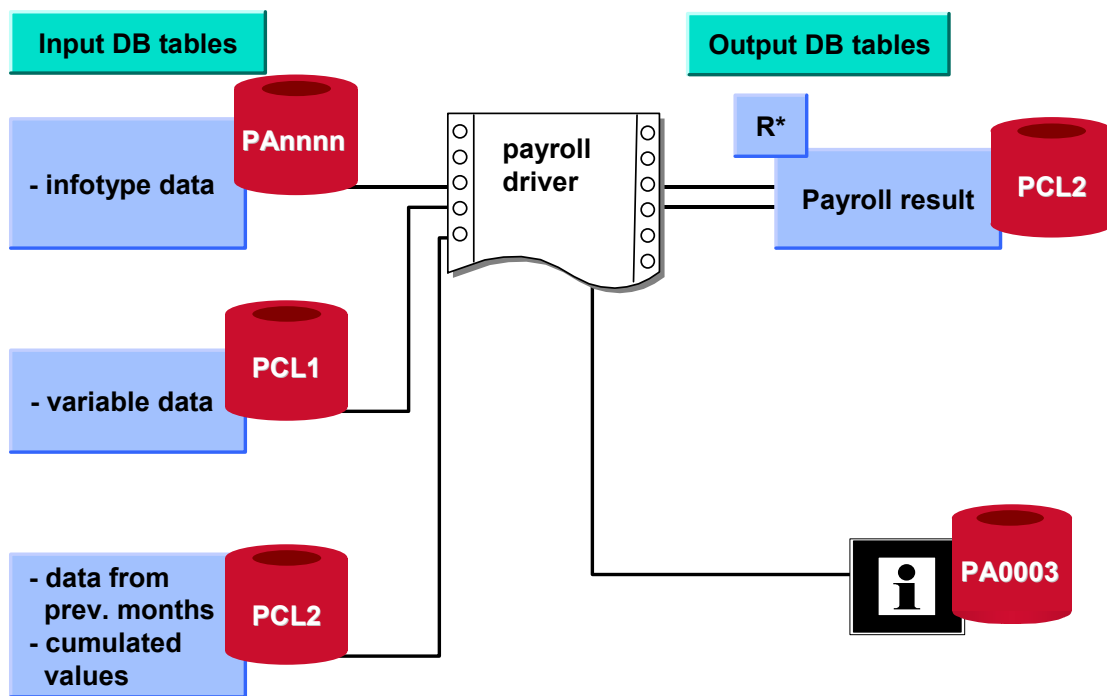
**Selection**

Personnel number: [ ] [ ]

Payroll area: U1 [ ]

© SAP AG 1999

- The payroll area is written directly from the control record to the payroll program. In a productive system, the payroll period is a default value.
- Personnel numbers are selected for the payroll run by entering a payroll area. The current payroll period is the last period accounted plus one.
- The control record contains the last period accounted for each payroll area.



© SAP AG 1999

- Personnel data such as Basic Pay (infotype 0008/master data) and Employee Remuneration Information (infotype 2010/time data) is imported to the payroll program from the infotypes (database table PAnnnn).
- The payroll program maintains the Payroll Status infotype (0003), including retroactive accounting, correction runs, matchcode W, and so on.
- PCL1 Primary data such as time tickets and time results.
- PCL2 Secondary data such as time pairs and payroll results.  
The cluster identifier for the country-specific payroll results is stored in table T500L.
- The name of the cluster for the country-specific payroll results is stored in table T500L.

## Cluster display: technical view



© SAP AG 1999

- The RPCLSTxy reports enable you to view different clusters in the database tables PCL1 or PCL2.
- PCL2 contains the following clusters that are relevant to payroll:
  - PS Schema
  - xy Payroll results
  - CU Directory of payroll results
- Cluster RU contains the results for the US payroll run and cluster RD contains the results of the payroll run for Germany.

