

HR350

Programming in Human Capital Management

mySAP ERP Human Capital Management

Date _____
Training Center _____
Instructors _____
Education Website _____

Participant Handbook

Course Version: 2005 Q2
Course Duration: 5 Day(s)
Material Number: 50073375



An SAP course - use it to learn, reference it for work

Copyright

Copyright © 2005 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. Additionally this publication and its contents are provided solely for your use, this publication and its contents may not be rented, transferred or sold without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Trademarks

- Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.
- IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.
- ORACLE® is a registered trademark of ORACLE Corporation.
- INFORMIX®-OnLine for SAP and INFORMIX® Dynamic Server™ are registered trademarks of Informix Software Incorporated.
- 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.
- SAP, SAP Logo, R/2, RIVA, R/3, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Disclaimer

THESE MATERIALS ARE PROVIDED BY SAP ON AN "AS IS" BASIS, AND SAP EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THESE MATERIALS AND THE SERVICE, INFORMATION, TEXT, GRAPHICS, LINKS, OR ANY OTHER MATERIALS AND PRODUCTS CONTAINED HEREIN. IN NO EVENT SHALL SAP BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES OF ANY KIND WHATSOEVER, INCLUDING WITHOUT LIMITATION LOST REVENUES OR LOST PROFITS, WHICH MAY RESULT FROM THE USE OF THESE MATERIALS OR INCLUDED SOFTWARE COMPONENTS.

About This Handbook

This handbook is intended to complement the instructor-led presentation of this course, and serve as a source of reference. It is not suitable for self-study.

Typographic Conventions

American English is the standard used in this handbook. The following typographic conventions are also used.

Type Style	Description
<i>Example text</i>	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths, and options. Also used for cross-references to other documentation both internal (in this documentation) and external (in other locations, such as SAPNet).
Example text	Emphasized words or phrases in body text, titles of graphics, and tables
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example SELECT and INCLUDE.
Example text	Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, and passages of the source text of a program.
Example text	Exact user entry. These are words and characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

Icons in Body Text

The following icons are used in this handbook.

Icon	Meaning
	For more information, tips, or background
	Note or further explanation of previous point
	Exception or caution
	Procedures
	Indicates that the item is displayed in the instructor's presentation.

Contents

Course Overview	vii
Course Goals	vii
Course Objectives	vii
Unit 1: Data Structures in Personnel Administration	1
Data Structures and Infotypes	2
Check and Control Tables	14
Unit 2: Logical Databases PNP/PNPCE	27
Overview of Logical Database PNP	28
Data Screening	37
Logical Database PNPCE	47
Report Categories	50
Unit 3: Infotypes and Function Modules	65
Joining Infotypes	66
Field Projection	73
Function Modules	81
Unit 4: Repetitive Structures and List Display	105
Infotypes with Repetitive Structures	106
List Display with ABAP List Viewer	111
Unit 5: Cluster Database Tables in HCM	129
Database Tables	130
Importing Data	136
Payroll Results	147
Unit 6: Data Structures in Personnel Planning	171
Functions and Data Structure	172
Personnel Planning Data Structures	180
Unit 7: Logical Database PCH	197
Structure Formats	198
Evaluations-Source Code	206

Unit 8: Customer-Specific Infotypes223

- Overview of Infotypes 224
- Personnel Administration Infotypes 230
- Personnel Planning Infotypes 250

Unit 9: HCM ABAP Features269

- Subroutine Calls 270
- Report Selection 278

Appendix 1: Report Categories and Clusters287

Appendix 2: Introduction to Human Capital Management295

Index327

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Course Overview

This course covers how to use HCM logical databases and the Join and Projection views. It outlines the use of HCM-specific statements. In addition, it explains how to import and process payroll results and create customer-specific infotypes.

Target Audience

This course is intended for the following audiences:

- IT staff
- IT administrators

Course Prerequisites

Required Knowledge

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



Course Goals

This course will prepare you to:

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



Course Objectives

After completing this course, you will be able to:

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

SAP Software Component Information

The information in this course pertains to the following SAP Software Components and releases:

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Unit 1

Data Structures in Personnel Administration

Unit Overview

This unit explains the technical data structures of the infotypes in Personnel Administration. In addition, it covers how to use the technical information in the check and control tables of HCM infotypes.



Unit Objectives

After completing this unit, you will be able to:

- Explain the structure of transparent tables
- Describe the fields in the structures
- Use the technical information contained in the check and control tables of HCM infotypes

Unit Contents

Lesson: Data Structures and Infotypes	2
Lesson: Check and Control Tables	14
Exercise 1: Data Structures in Personnel Administration	19

Lesson: Data Structures and Infotypes

Lesson Overview

This lesson explains the technical data structures of the infotypes in Personnel Administration and their key fields.



Lesson Objectives

After completing this lesson, you will be able to:

- Explain the structure of transparent tables
- Describe the fields in the structures

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. As a member of the IT team, you have to make optimal use of the three include structures of the transparent database tables, PAnnnn. You also have to add customer-specific fields to Personnel administration infotypes.

Infotypes in Personnel Administration

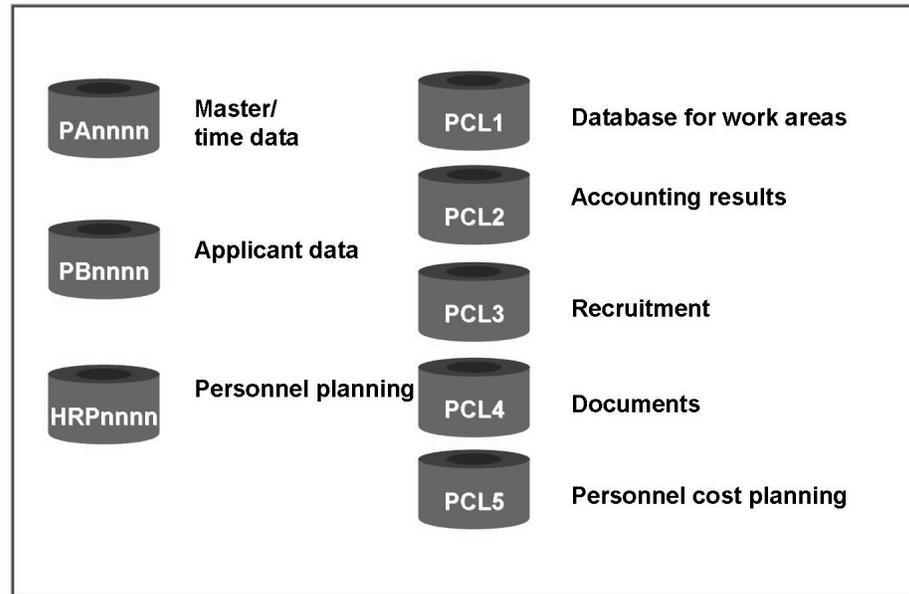


Figure 1: Data Structures in Context

Data structures in HCM 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 the cluster, PCL1, or accounting results on the cluster, PCL2.

The infotypes are grouped into different **number ranges, nnnn:**

- 0000 – 0999 Personnel Administration master data
- 1000 – 1999 Personnel Planning
- 2000 – 2999 Time Management
- 4000 – 4999 Recruitment
- 9000 – 9999 Customer-specific enhancements

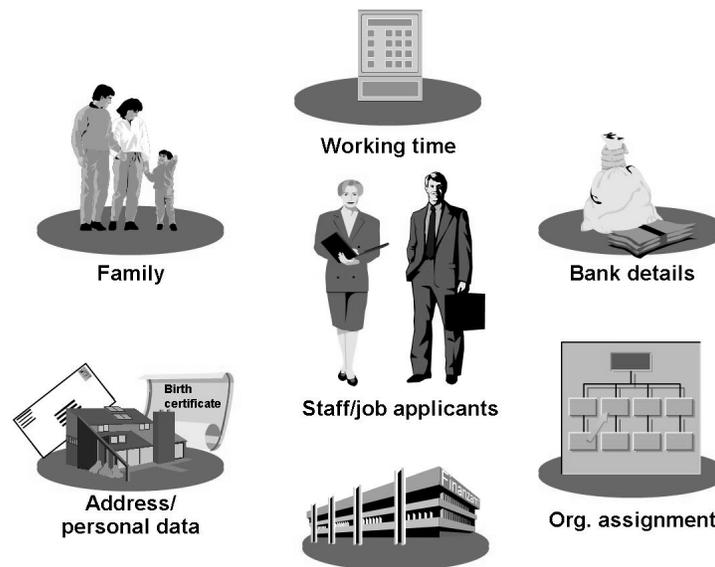


Figure 2: Infotypes in Personnel Administration

Infotypes are information units in the Human Capital 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 the prerequisites to enter 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. As a result, several data records normally 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** that break down information further. For example, the address types for the infotype *Address* 0006 represent the subtypes.

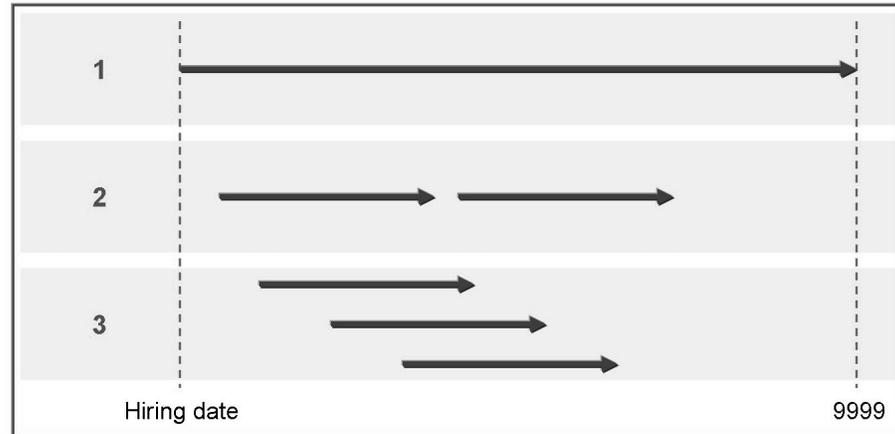


Figure 3: Infotype Time Constraint

Each infotype and subtype has a time constraint.

Time constraint 1: Complete saving with no overlaps at any time, which means one valid data record of the relevant infotype must exist during the duration of the employee's tenure at the company.

Time constraint 2: Incomplete saving without overlaps at any time, which means 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, which means any number of valid data records of an infotype may exist at any given time.

Technical Data Structures of Infotypes



The structure of transparent tables:

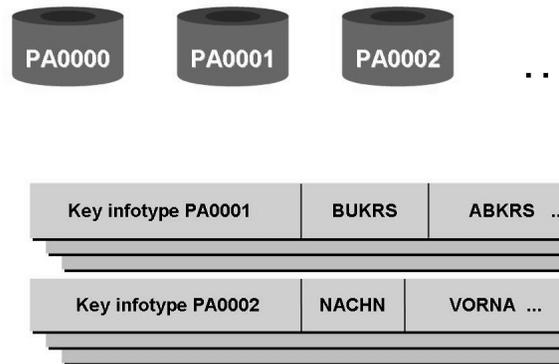


Figure 4: Structure of Transparent Tables

The data records of the infotype, **nnnn**, are saved in the transparent database tables, **PAxxxx**.

The table's **primary key** indicates a minimum quantity of fields in a table, the value of which 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 do not uniquely identify the data records.

The fields involved in a primary key are called the table's **key fields**. This means that a value combination of the key fields may only occur one time in a table.

A **transparent table** is made up of the primary key and other nonprimary keys. In case of infotype tables, nonprimary fields include the infotype-specific fields that contain the actual data or information on the infotype.



- Transparent tables *dictionary field definition*

Example: PA0002

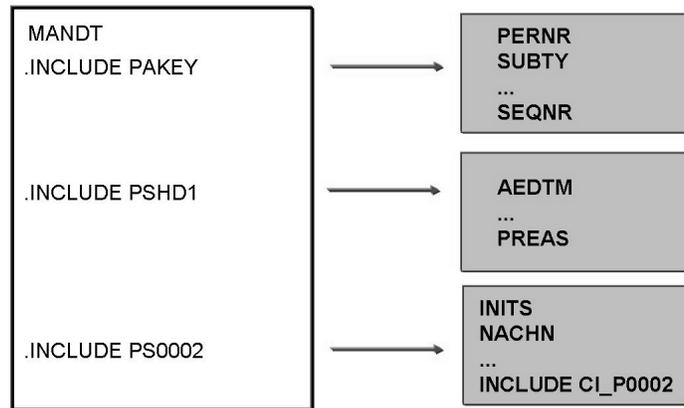


Figure 5: Personnel Administration Transparent Tables

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 about the infotype record, which includes 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 actual information and data of the infotype.

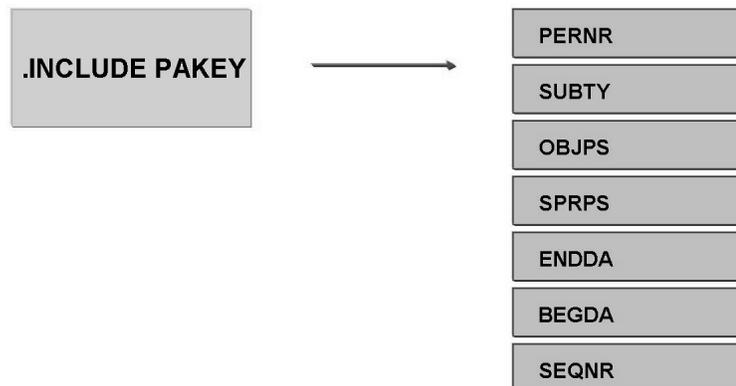


Figure 6: Key Fields

The **PAKEY** structure contains the primary key of an infotype. It is the same for all Personnel Administration infotypes and is made up of the following key fields:

- **PERNR**: Represents the personnel number, which is the only unique key that identifies an employee in a client. It can be used to display and maintain master data and working time data (infotypes) for an employee.
- **SUBTY**: Signifies the subdivisions of infotypes. The subtypes of an infotype may have different time constraints and create their own history.
- **OBJPS**: Differentiates records with the same infotype, subtype, lock indicator, valid from date, and valid to date. For example, the Child number in the infotype, *0021 Family Member/Dependents*.
- **SPRPS**: Represents the lock indicator for HCM master data. It can be used to lock and unlock data records. It can also be used to ensure double control, which means a minimum of 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 the infotype records with the same key and the time constraint, 3. In contrast to the personnel object identification, this number is automatically assigned by the system.

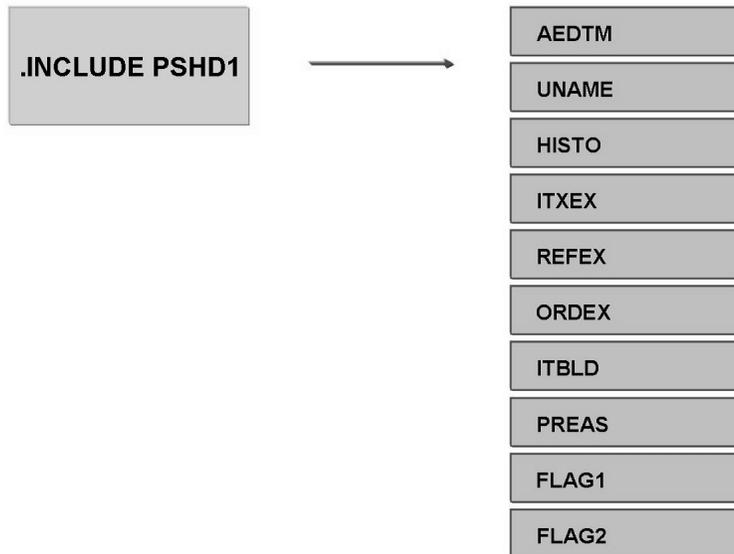


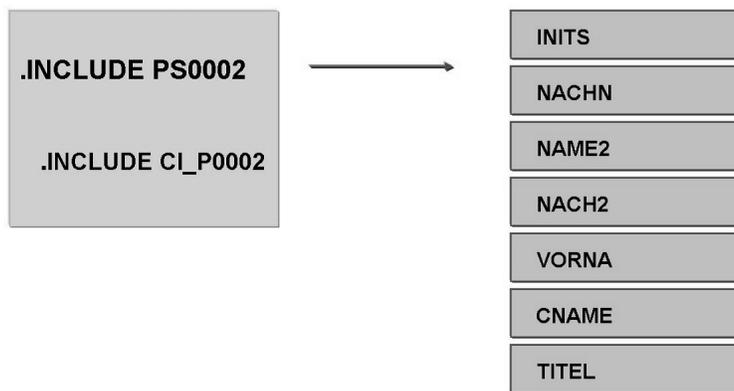
Figure 7: Administration Fields

The structure, **PSHD1**, is another basic structure that is the same for all Personnel Administration infotypes. The 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 about whether the administrator created plain text for this infotype (ITXEX).



Example: PA0002



...

Figure 8: Data Fields in the Structure PSnnnn

The structure, **PSnnnn**, contains the infotype-specific data fields, which means 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.

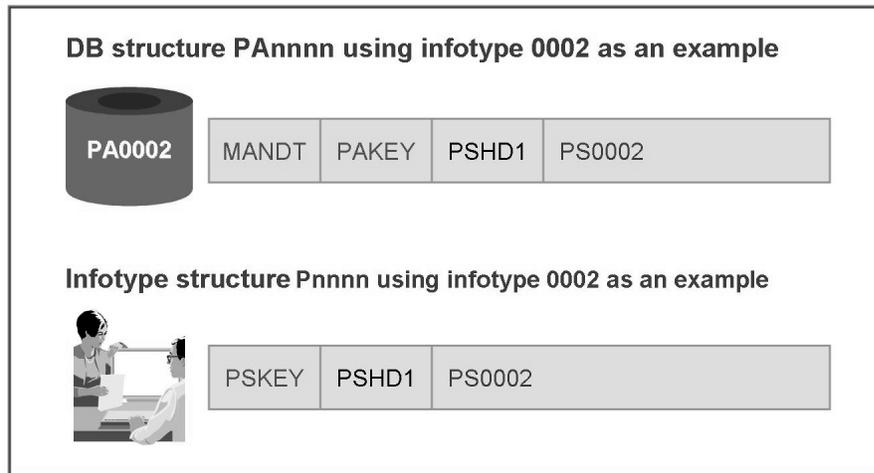


Figure 9: The Infotype Structure, Pnnnn

During the interface definition, the logical structure, **Pnnnn**, is used between programs and among 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.

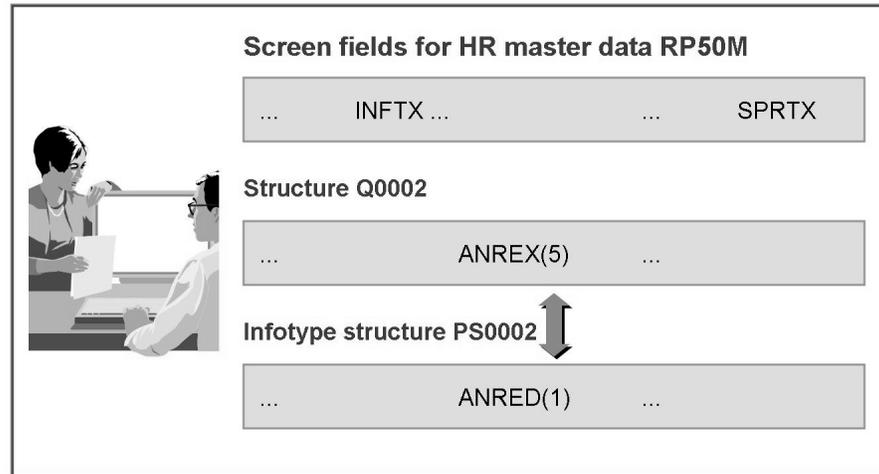


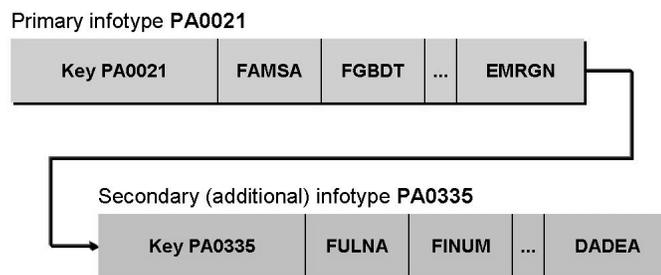
Figure 10: Additional Structures for Screen Fields

The fields, such as header lines, which are displayed for all infotypes are stored in the structure, RP50M.

The structure, **Qnnnn**, also exists for several infotypes. It contains the screen fields for data entry. For example, an employee’s form of address, is stored with a numeric key in the field, PA0002-ANRED. Notice that the screen must enable the form of address itself to be displayed and entered, and not its numeric key. For this reason, the additional field, Q0002-ANREX, is used.



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



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

Figure 11: Infotype Views/Additional Infotypes

Many country versions require country-specific fields that are attached to all international infotypes. Examples of such fields are *Organizational Assignment* and *Family Member/Dependents*.

To prevent international infotype structures from being overloaded with country-specific fields, the fields have been moved to **additional infotypes**.

International infotypes (“primary infotype”) have the value, *I*. As opposed to this, 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**, which is defined using the tables, T582V and T582W. Next, the infotypes are saved, and the data you are maintaining is distributed across two infotype records that consist of one primary infotype record and one additional infotype record, and have an identical infotype key.

The infotype menu only contains the international primary infotype. The additional infotype can be maintained only with the primary infotype, and not on its own. All infotype characteristics such as the time constraint are defined only 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, which must not be changed.



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

Figure 12: Overview of PA Data Structures

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



Lesson Summary

You should now be able to:

- Explain the structure of transparent tables
- Describe the fields in the structures

Lesson: Check and Control Tables

Lesson Overview

This lesson describes the technical information in the check and control tables of HCM infotypes. It covers the types of applicant infotypes and the subtypes of Personnel Administration infotypes.



Lesson Objectives

After completing this lesson, you will be able to:

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

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. As a member of the IT team, you have to make a check on the data entered in HCM infotypes. You also have to keep a check on the new infotypes created.

Check and Control Tables

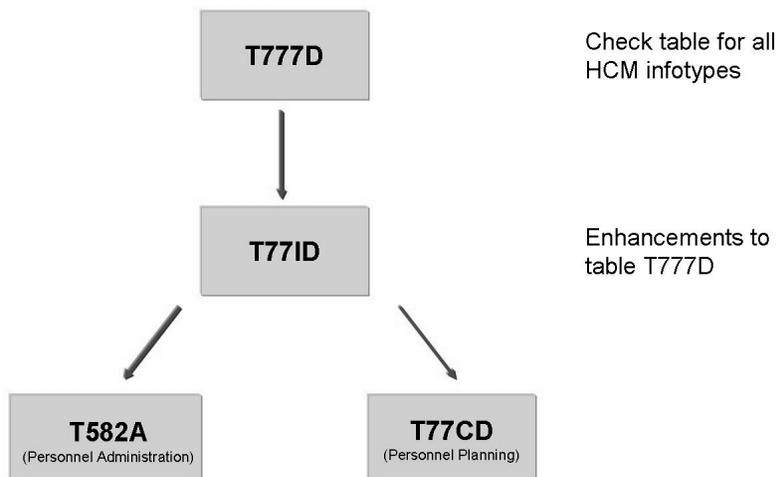


Figure 13: Check and Control Tables HCM Infotypes

The table, *Infotypes - Dialog/Database Assignment (T777D)*, is the key check table (domain INFOTYP) for all HCM infotypes, including 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, which 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 to plan infotypes.

You can use the report, RHT777DCHECK, to check the consistency of the 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	

Figure 14: Check Table HCM Infotypes

The table, *Infotypes - Dialog/Database Assignment (T777D)*, is automatically maintained using the tool to create new infotypes. Do not edit this table manually.

In addition to the transparent tables, **PA#####** and **HRP#####**, each infotype in the Human Capital Management system has a **logical structure, P#####** (stored in T777D-PPNNN).

You use logical structures to define the interfaces between programs and among screens and programs. No data records exist on the database for a logical structure.

To avoid direct database accesses, the logical structures, P#####, are primarily used in Human Resources Management programs.



Table T77ID

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	

Figure 15: Infotypes - Enhancements to T777D

The table, *Infotypes - Enhancements to T777D* (T77ID), supplements only the *Infotypes - Dialog/Database Assignment* table, T777D, for reasons of memory space.

This table includes the structure, PSnnnn or HRIInnnn, for the infotype-specific fields for each infotype.



- Control applicant infotypes

Table T777D

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

Figure 16: Database Tables for Applicant Infotypes

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. In this case, there is a database table, **PA#####**, for employee data and a database table, **PB#####**, for applicant data.
- Infotypes that may be maintained only for applicants (applicant infotypes only, with infotype numbers from 4000 to 4999). In this case, there is only one database table solely meant for applicant data, **PB4###**.



- **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	

Figure 17: Subtypes for Personnel Administration Infotypes

Subtypes are used to break down *infotypes*, such as a subtype to further classify the infotype, 0006 *Addresses*, into main address, second address, and business address.

In the table, T777D, the following fields define subtypes:

- **Subtype field:** Represents the field that is in the infotype structure, which contains the subtype. Normally this is the field, *SUBTY*. Notice that it may also be a field other than *SUBTY* and contain the value of the subtype, such as *ANSSA* = address type in the infotype, 0006.
- **Subtype table:** Contains the check table that consists of all the subtype values allowed for each infotype. This is normally the table, *T591A*. Notice that the subtype values may also be stored in another table depending on the subtype field, such as the table, *T512Z*, for the infotype, 0011 *Ext. Bank Transfers* (permissible wage types for each infotype), as the wage type, (field *LGART*), is also the subtype.
- **Subtype text table:** Lists the text table for the subtype table, which is the table with the names for the individual subtype values. This is normally the table, *T591S*, if the table, *T591A*, is used as a subtype table. If another subtype table is used, (such as *T512Z*), the corresponding text table, such as *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*).

Exercise 1: Data Structures in Personnel Administration

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. As you are the HCM manager, you want to draw up your own evaluations of Personnel Planning data. To do this, you must learn more about data structures in Personnel Administration.

Task 1:

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

1. **SPRPS:**
2. **OBJPS:**

Task 2:

What tasks do the structures, Qnnnn and CI_Pnnnn, complete?

1. **Qnnnn:**
2. **CI_Pnnnn:**

Task 3:

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

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

Solution 1: Data Structures in Personnel Administration

Task 1:

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

1. **SPRPS:**

a) **SPRPS:**

If this field is not blank, the infotype record is locked. You need to unlock the infotype record to activate it.

2. **OBJPS:**

a) **OBJPS:**

This field is used to differentiate the infotype records with the same infotype, subtype, block indicator, valid from date, and valid to date.

Task 2:

What tasks do the structures, Qnnnn and CI_Pnnnn, complete?

1. **Qnnnn:**

a) **Qnnnn:**

The structure, Qnnnn, contains screen fields with different lengths than the corresponding database fields.

2. **CI_Pnnnn:**

a) **CI_Pnnnn:**

The customer-specific fields of a standard infotype are grouped in the structure, CI_Pnnnn.

Continued on next page

Task 3:

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

1. How do you determine which time constraint is assigned to the subtype of an infotype?
 - a) To establish the time constraint of the subtype of the infotype, determine the time constraint table that is assigned to the infotype in the table, T777D. View the table.



Lesson Summary

You should now be able to:

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



Unit Summary

You should now be able to:

- Explain the structure of transparent tables
- Describe the fields in the structures
- Use the technical information contained in the check and control tables of HCM infotypes

Internal Use SAP Partner Only

Internal Use SAP Partner Only



Test Your Knowledge

1. From a database point of view, infotypes provide a _____ and a set of coherent data records.

Fill in the blanks to complete the sentence.

2. A _____ is made up of the primary key and other nonprimary keys.

Fill in the blanks to complete the sentence.

3. How can customer specific fields be added to the Personnel Administration infotype?

4. The table, Infotypes - Dialog/Database Assignment (T777D), cannot be automatically maintained.

Determine whether this statement is true or false.

- True
 False

5. _____ are used to further break down infotypes.

Fill in the blanks to complete the sentence.



Answers

1. From a database point of view, infotypes provide a data structure and a set of coherent data records.

Answer: data structure

2. A transparent table is made up of the primary key and other nonprimary keys.

Answer: transparent table

3. How can customer specific fields be added to the Personnel Administration infotype?

Answer: The 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.

4. The table, Infotypes - Dialog/Database Assignment (T777D), cannot be automatically maintained.

Answer: False

The table, Infotypes - Dialog/Database Assignment (T777D), is automatically maintained using the tool to create new infotypes. Do not edit this table manually.

5. Subtypes are used to further break down infotypes.

Answer: Subtypes

Unit 2

Logical Databases PNP/PNPCE

Unit Overview

This unit enables you to use the logical database, PNPCE, in the PNP mode. It also describes how to program the selection screen of the logical database to suit customer-specific needs.



Unit Objectives

After completing this unit, you will be able to:

- List the functions of the logical database PNP
- Retrieve period-based data
- Use the screening criteria to select data and persons
- Sort data according to the organizational criteria
- Use the logical database, PNPCE, in the PNP mode
- Create and assign report categories
- Create a selection view

Unit Contents

Lesson: Overview of Logical Database PNP.....	28
Lesson: Data Screening	37
Exercise 2: Creating a List of Employees	43
Lesson: Logical Database PNPCE	47
Lesson: Report Categories	50
Exercise 3: Creating a Customer Report Class	55

Lesson: Overview of Logical Database PNP

Lesson Overview

This lesson provides an overview of the functions of the logical database, PNP. It explains how to process infotypes. In addition, it covers how to retrieve data for specific periods according to your requirements.



Lesson Objectives

After completing this lesson, you will be able to:

- List the functions of the logical database PNP
- Retrieve period-based data

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The executive board needs to create HCM reports using the logical database, PNP.

Functions of Logical Database PNP

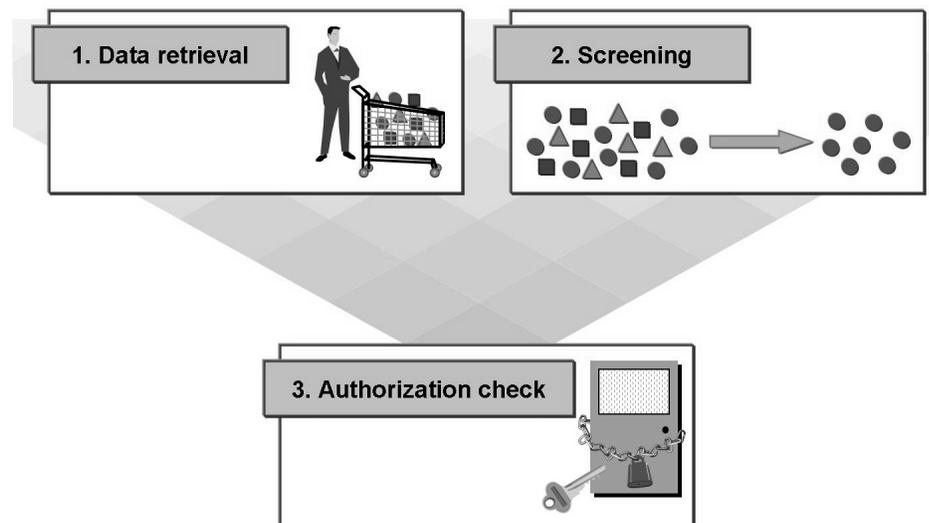


Figure 18: Functions of the Logical Database PNP

To minimize the amount of programming required, it is a good idea to use logical databases when you generate 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: HCM 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 the organizational criteria. For example, you can select all 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 is not fruitful to allow all programs to carry out the authorization check.

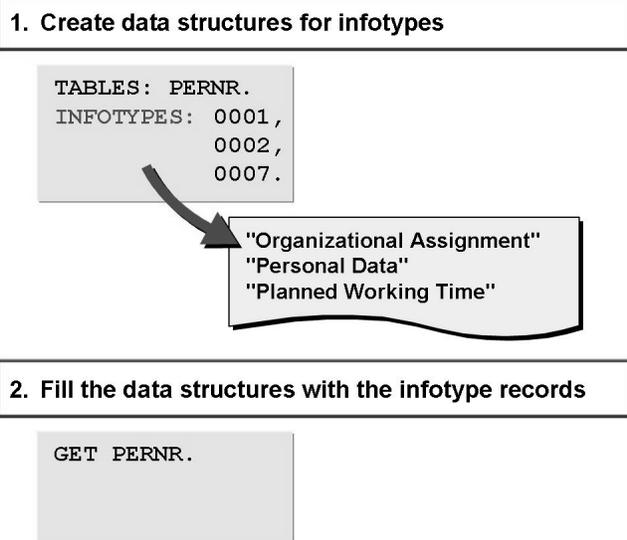


Figure 19: Data Retrieval

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 the declared infotypes are filled with all the records that exist for a personnel number. The field, **PERNR**, in the structure, **PERNR**, contains the personnel number. For 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. For this reason, declare only the infotypes you want to process. You can use the enhancement **MODE N**, for the statement, **INFOTYPES**, to suppress the filling of internal tables.

Processing Infotypes



```

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.

```

Figure 20: Processing Infotypes (1)

When the GET PERNR event occurs, the infotype records are imported to the internal tables, Pnnnn, such as P0006 for the infotype, 0006. The tables may then be processed in a **LOOP-ENDLOOP** loop.

All the infotype records, the validity period of which 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 select 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.

```

Figure 21: Processing Infotypes (2)

Infotypes with the time constraints, 1 or 2, may be processed with a **PROVIDE-ENDPROVIDE** loop.

All infotype records, the validity period of which 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.

```

Figure 22: Processing Infotypes (3)

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, the processing of the infotype, *Addresses* (0006), is restricted to the subtype, 1 *Permanent Address*. This prevents the intervals from overlapping. This is required as the PROVIDE processing occurs correctly only for the infotype records with the 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.

```

Figure 23: Processing Infotypes (4)

Frequently, you only want to display employees with the 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...
  ENDLLOOP.
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.

```

Diagram illustrating loop nesting:

- The first loop (LOOP AT P0002) is labeled as an **Infotype loop**.
- The second loop (PROVIDE * FROM P0002) is labeled as an **Infotype loop**.
- The third loop (PROVIDE * FROM P0006) is labeled as an **Infotype loop**.
- The entire structure is labeled as an **Employee loop**.

Figure 24: Loop Nesting

HCM data is processed in two nested loops:

A GET PERNR loop for all the personnel numbers selected. It is concluded implicitly by the next event, such as END-OF-SELECTION.

Subordinate loops for each 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.

Period-Based Data

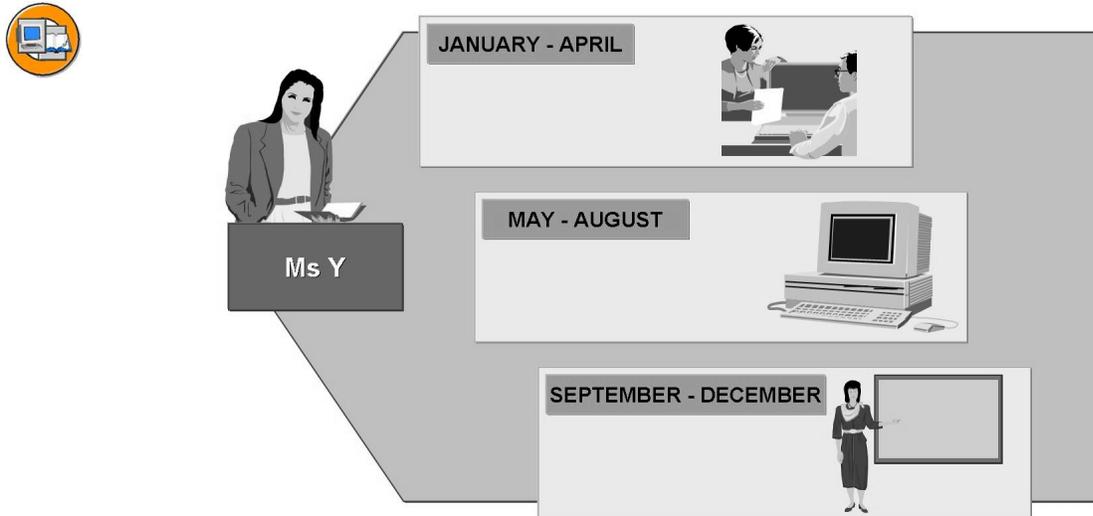


Figure 25: Period-Based Data (1)

Infotype data is period-based, which means it is valid only 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.

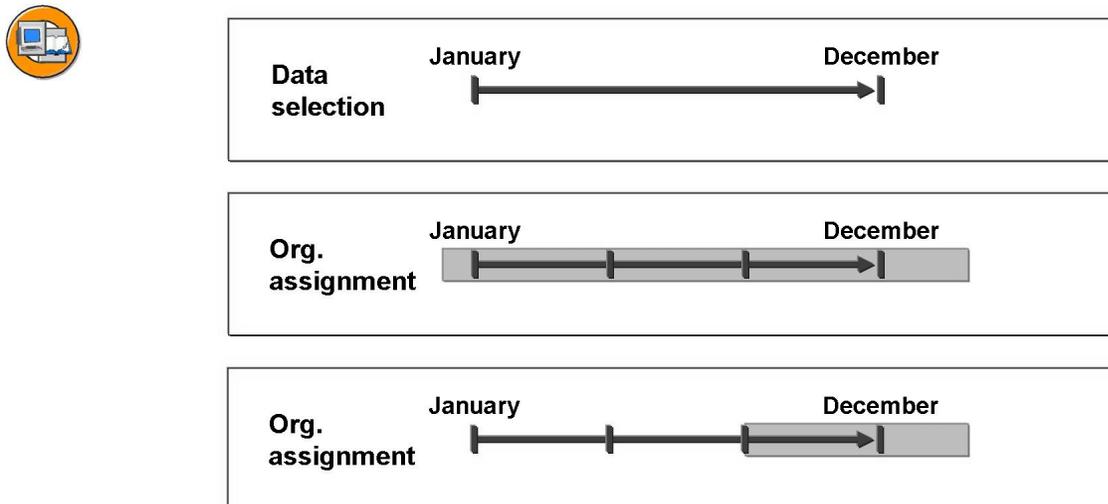


Figure 26: Period-Based Data (2)

Both types of data selection are based on the data selection period entered in the selection screen. Processing with the LOOP-ENDLOOP and PROVIDE-ENDPROVIDE provides the data for a specific period, which means the processing provides data according to the data selection period.

Frequently, only the most recent or oldest infotype record is required from the selected period, instead of all the infotype records. The RP_PROVIDE_FROM_LAST and RP_PROVIDE_FROM_FRST macros can be used in such situations.



Lesson Summary

You should now be able to:

- List the functions of the logical database PNP
- Retrieve period-based data

Lesson: Data Screening

Lesson Overview

This lesson shows how to screen HCM data. It also explains how to sort, process, and retrieve data for groups of employees for whom evaluations have to be conducted.



Lesson Objectives

After completing this lesson, you will be able to:

- Use the screening criteria to select data and persons
- Sort data according to the organizational criteria

Business Example

The Training International company has active employees, retirees, contractors, and external employees. The personnel department of the company needs to create a report for a group of employees for whom an evaluation is to be created.

Screening Criteria

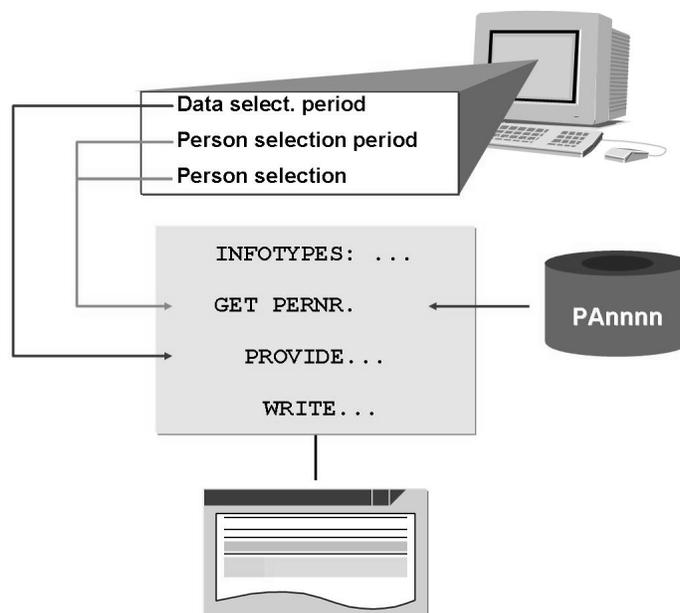


Figure 27: Screening Criteria

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 the employees who correspond to the selection criteria, such as 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 HCM records.



```

GET PERNR.

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

ENDPROVIDE.

```

Figure 28: Date and Person Selection Period

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 to select the personnel numbers to be processed. All the personnel numbers that comply with the selection criteria specified 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 above, *Today* is set as the key date. In this case, the four fields contain the system date.

Notice that all of the data records for an infotype are available in the corresponding internal table for GET PERNR. This means that data selection does not restrict the ability to read from the database.

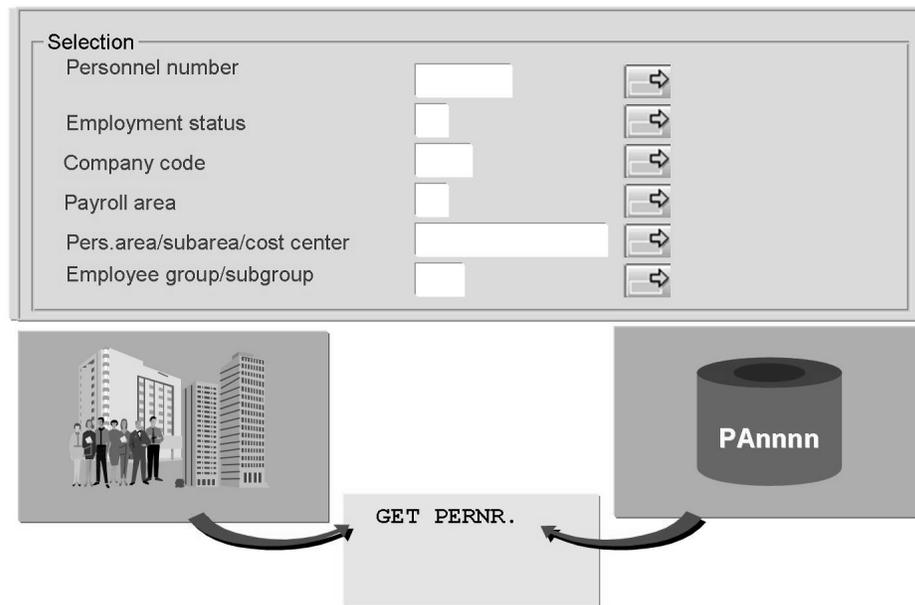


Figure 29: Person Selection

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, such as a range of personnel numbers, or the characteristics of the organizational assignment, or the status of the employee as to whether the employee 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 made available to your report only when the event, GET PERNR, occurs if these criteria are fulfilled.



Period		
<input type="radio"/> Today	<input type="radio"/> Current month	<input checked="" type="radio"/> Current year
<input type="radio"/> To current date	<input type="radio"/> From today	
<input type="radio"/> Other period	<input type="text"/>	to <input type="text"/>
Selection		
Personnel number	<input type="text"/>	<input type="button" value="→"/>
Payroll area	<input type="text"/>	<input type="button" value="→"/>

```

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

```

Figure 30: Default Values in the Selection Screen

When the INITIALIZATION event occurs, you can set default values for the selection screen.

The default radio buttons to select 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 the past to the 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.

Sort Order

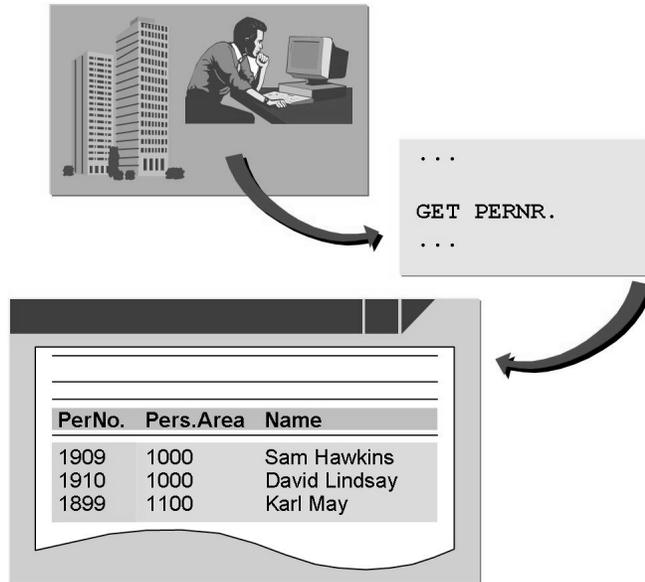


Figure 31: Sort Order

The standard sort sequence is in ascending order by the personnel number.

You can use the sort function to sort an evaluation using the organizational criteria.

For example, you can use the sort function to create a hierarchical list according to the personnel subarea and administrator, or list the employees in alphabetical order.

The sort function can be used for all of the fields in the 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 the matchcode sequence.

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Exercise 2: Creating a List of Employees

Exercise Objectives

After completing this exercise, you will be able to:

- Process infotype records using PROVIDE...ENDPROVIDE

Business Example

The personnel department in your company requires a list of employees, which indicates the language and the date of birth.

Task:

Create a report that generates a list of employees with the following information:

Personnel number

Name

Language

Date of birth

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

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

Take the formatted names from the field, **PERNR-ENAME**.

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



Hint: Create your report using the name, **ZPG##Enr**.

= Group number

nr = Exercise number (start with 01)

Solution 2: Creating a List of Employees

Task:

Create a report that generates a list of employees with the following information:

Personnel number

Name

Language

Date of birth

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

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

Take the formatted names from the field, **PERNR-ENAME**.

Continued on next page

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



Hint: Create your report using the name, **ZPG##Enr**.

= Group number

nr = Exercise number (start with 01)

```
a)  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-endda.
            CHECK language.
            WRITE: / p0002-pernr,
                   sy-vline,
                   pernr-ename,
                   sy-vline,
                   p0002-sprsl,
                   sy-vline,
                   p0002-gbdat.
        ENDPROVIDE.
```



Lesson Summary

You should now be able to:

- Use the screening criteria to select data and persons
- Sort data according to the organizational criteria

Lesson: Logical Database PNPCE

Lesson Overview

This lesson describes how to use the logical database, PNPCE, in the PNP mode. It also explains how to use the concurrent employment function.



Lesson Objectives

After completing this lesson, you will be able to:

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

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. As a member of the IT team, you have to make optimal use of the concurrent employment function offered by the logical database, PNPCE.

Logical Database PNPCE



- Employee as a person with multiple employments 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

Since the mySAP ERP 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 the infotype, 0709 *Person ID*.

To enable the concurrent employment function, you must use the logical database, PNPCE. Notice that when you use new developments in the mySAP ERP Release and do not use the concurrent employment function, it is a good idea to use this new logical database and improved selection screen. This is because the features allow you to use more extensive 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.

```

Figure 32: The Logical Database PNPCE in PNP Mode

In the report, the structure, PERNR, must be declared by the TABLES statement. Notice that any further use of the structure, PERNR, is restricted. For example, the event, GET PERNR, is not permitted. Instead, you use **GET PERAS**. Except for the component, PERNR-PERNR, all the other components of the structure, PERNR, are not populated any more and have initial values. For this reason, programming on these values, except PERNR-PERNR, is not permitted.

The events, GET GROUP and GET PERSON, are also available. When the events occur, you must declare them using NODES. If the events do not occur, it is assumed that the concurrent employment function is not required. The report then runs in the PNP mode.

Only a few companies are currently using the concurrent employment function, and for this reason, more information is not provided in this section. Note: For more information about using the concurrent employment mode, see the documentation for the program, SAPDBPNPCE.



Lesson Summary

You should now be able to:

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

Lesson: Report Categories

Lesson Overview

This lesson describes how to create and assign report categories. It also explains how to create a selection view.



Lesson Objectives

After completing this lesson, you will be able to:

- Create and assign report categories
- Create a selection view

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. As a member of the IT team, you have to program the selection screen of the logical database to suit customer-specific needs.

Creating and Assigning Report Categories



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	Person selection period	Payroll area/period/year
<input type="radio"/> No entry	<input type="radio"/> No entry	<input checked="" type="radio"/> No entry
<input type="radio"/> Key date	<input type="radio"/> Key date	<input type="radio"/> Entry possible
<input checked="" type="radio"/> Interval	<input checked="" type="radio"/> Interval	<input type="radio"/> Mandatory field

Type SAP

Name PNP_DEFAULT

Figure 33: IMG: Create Report Categories

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 through 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. In addition, indicate 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.

To select the required selection fields, double-click *Permitted Selection Criteria* in the *Dialog Structure* group box. Select the fields that should appear on the first page of the Selection screen. The fields that are not selected are displayed in an additional window when you select *Further Selections*.

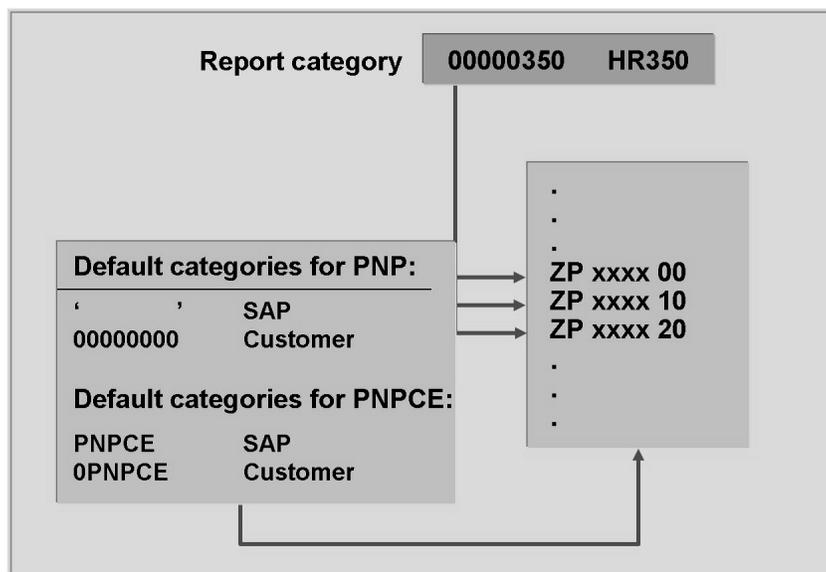


Figure 34: IMG: Assign Report Categories

In the standard SAP system, a report class is assigned to the reports. If you want to override this class, carry out the steps described here. Making an assignment to report the classes for standard reports using SE38 constitutes a system modification.

In case of the reports that you have developed, you can make the assignment to report classes using either the Customizing activity outlined here, (client-specific), or the report attributes in SE38 (for all clients).

If a report uses the logical database, PNP, you can access the HCM report category maintenance functions from the *Program Attributes*. To do this, select *Edit* → *HR Report Category*. The report category currently assigned to the report appears 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 a 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.

Creating a Selection View



1.

Create Selection View

Origin of view

For logical database

For any tables

Name of view

2.

Tables

PA0002
PA0006

Figure 35: Creating a Selection View (1)

Free delimitation is one option you can use to add the selection criteria to the Selection screen. When you use this option, the additional selections are made directly by the logical database, and for this reason, the performance is 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, select *Edit object*. Under *More...*, click **Selection view** and choose *Create*.

Enter the names of the database tables, the fields of which you want to use for the selection view, on the next screen. 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	

Figure 36: Creating a Selection View (2)

Within a selection view, fields are selected and grouped together as **functional groups**. All the fields assigned to a functional group are included in the view. By grouping the fields into functional groups, you can keep the fields with similar contents together. This enables you to use the view to search for specific information.

To define a functional group, assign a functional group identifier and a corresponding text. The function group identifier can consist of any two characters and is only relevant to maintain the selection view.

From the *Table/Node* window, select the table, the fields of which you want to assign to one of your defined functional groups. The system lists the fields on the *Fields in Table/Node* screen from where you can assign the required fields to a functional group.

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Exercise 3: Creating a Customer Report Class

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. As a member of the IT team, the personnel department wants you to program the selection screens for the customer reports in such a way that they display the selection fields required for the corresponding reports.

Task 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

Task 2:

1. Assign the new report class to your report.

Solution 3: Creating a Customer Report Class

Task 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

- a) 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.

Select your new report class and select *Allowable Selection Criteria* in the dialog structure, on the Overview screen. 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.

Continued on next page

Task 2:

1. Assign the new report class to your report.
 - a) 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.

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



Lesson Summary

You should now be able to:

- Create and assign report categories
- Create a selection view



Unit Summary

You should now be able to:

- List the functions of the logical database PNP
- Retrieve period-based data
- Use the screening criteria to select data and persons
- Sort data according to the organizational criteria
- Use the logical database, PNPCE, in the PNP mode
- Create and assign report categories
- Create a selection view

Internal Use SAP Partner Only

Internal Use SAP Partner Only



Test Your Knowledge

1. Processing with LOOP-ENDLOOP and PROVIDE-ENDPROVIDE provides the data for a specific period, which means the processing provides data according to the data selection period.

Determine whether this statement is true or false.

- True
- False

2. The _____ logical database PNP function enables you to select employees according to the organizational criteria.

Fill in the blanks to complete the sentence.

3. The sort function enables you to sort an evaluation in accordance with the organizational criteria.

Determine whether this statement is true or false.

- True
- False

4. Fields for the data selection are defined in the structure, QPPNP. What are the possible values for the field, PNPTIMED?

5. When the events, GET GROUP and GET PERSON, occur, you must declare them using _____.

Fill in the blanks to complete the sentence.

6. What needs to be used to enable the concurrent employment function?

- 7. _____ is an option that you can use to add the selection criteria to the Selection screen.

Fill in the blanks to complete the sentence.

- 8. How can you make the assignment to report classes for reports that you have developed?



Answers

1. Processing with LOOP-ENDLOOP and PROVIDE-ENDPROVIDE provides the data for a specific period, which means the processing provides data according to the data selection period.

Answer: True

Processing with LOOP-ENDLOOP and PROVIDE-ENDPROVIDE provides the data for a specific period, which means the processing provides data according to the data selection period.

2. The screening logical database PNP function enables you to select employees according to the organizational criteria.

Answer: screening

3. The sort function enables you to sort an evaluation in accordance with the organizational criteria.

Answer: True

The sort function enables you to sort an evaluation in accordance with the organizational criteria. For example, you can use the sort function to create a hierarchical list according to the personnel subarea and the administrator, or list employees in alphabetical order.

4. Fields for the data selection are defined in the structure, QPPNP. What are the possible values for the field, PNPTIMED?

Answer: The possible values for the field, PNPTIMED, are:

- 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)

5. When the events, GET GROUP and GET PERSON, occur, you must declare them using NODES.

Answer: NODES

6. What needs to be used to enable the concurrent employment function?

Answer: To enable the concurrent employment function, you must use the logical database, PNPCE.

7. Free delimitation is an option that you can use to add the selection criteria to the Selection screen.

Answer: Free delimitation

8. How can you make the assignment to report classes for reports that you have developed?

Answer: For reports that you have developed, you can make the assignment to report classes using either the Customizing activity (client-specific) or the report attributes in SE38 (for all clients).

Unit 3

Infotypes and Function Modules

Unit Overview

In this unit, you learn how to process two or more infotypes in a PROVIDE...ENDPROVIDE loop. When you process an infotype, you can use a projection for changes to the content of one or more fields. You can then link both the views and read the time-dependent text entries from the tables. You can also use the HCM-specific macros and function modules in your programs.



Unit Objectives

After completing this unit, you will be able to:

- Understand join and intervals
- Join two or more infotypes in a single PROVIDE - ENDPROVIDE loop
- Process one or more infotypes
- Combine the logical views of infotype data - join and projection
- Read time-dependent table entries
- Use macros
- Process infotype records
- Read infotypes without the logical database
- Update infotype records

Unit Contents

Lesson: Joining Infotypes.....	66
Exercise 4: Joining Infotypes	69
Lesson: Field Projection	73
Exercise 5: Projection to Infotype Fields	77
Lesson: Function Modules.....	81
Exercise 6: Macros.....	93

Lesson: Joining Infotypes

Lesson Overview

This lesson shows you how to process two or more infotypes simultaneously. As HCM data changes over time, you also learn how to generate meaningful data while processing two or more infotypes from different periods.



Lesson Objectives

After completing this lesson, you will be able to:

- Understand join and intervals
- Join two or more infotypes in a single PROVIDE - ENDPROVIDE loop

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The personnel department requires reports that combine data from different infotypes.

Introduction to Joins

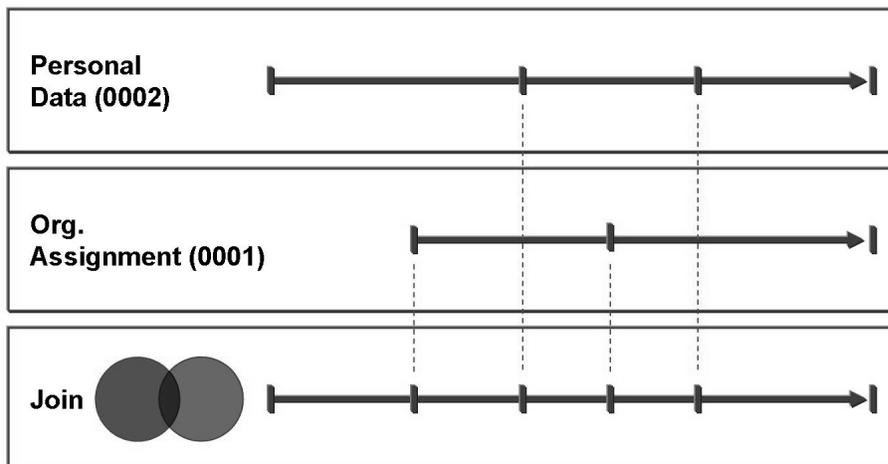


Figure 37: Join/Creating Intervals

Evaluations can be coded for specific infotypes, which means that each infotype is processed in its own PROVIDE loop. Alternatively, evaluations can be coded for all infotypes, which means that a single PROVIDE loop retrieves and processes the data from two or more infotypes at the same time. The infotypes to be processed are counted as a data source.

All HCM data changes over time. For this reason, time-dependencies must be taken into account when infotype data is linked by a JOIN. The validity of HCM data is not absolute. Instead, HCM data is valid only for specific periods. For this reason, a join creates one or more validity intervals in which the data for both the periods is valid.

If an evaluation is run for key dates, the data that is currently valid is retrieved in a data record for both the infotypes.

In principle, a join is a logical database operation performed on the time axis. New periods are created with the valid data using the specified validity periods of the infotypes to be linked. This means that new infotype records are created. This enables you to see the time-based inter-relationships among 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.

```

Figure 38: Join/Source Code

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 the **time constraints 1 or 2**.

The logical validity is stored in the BEGDA and ENDDA fields of the infotype.

Notice that changing the 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.

```

Figure 39: Join/Subtypes

If the 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 contains only personal data. The record is not meaningful because the task of the join to retrieve data from all the infotypes in question has not been performed.

Using the variable, Pnnnn_VALID, the system recognizes that one partial interval contains only incomplete data. When the report is run, this variable is created for each Pnnnn infotype included in a join. If a partial interval for the infotype, Pnnnn, contains data, its Pnnnn_VALID variable is filled with X.

Exercise 4: Joining Infotypes

Exercise Objectives

After completing 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

Business Example

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

Task:

Create a report that documents the period in which the employee lived at which address under which name. The created list should contain the following information:

Personnel number

Start date

End date

Name

Place of residence

1. Process the records from the infotype, 0006, only which have the subtype 1.

Print data only for the periods for which address information is available.

Use a variable with the characteristics of the field, **PERNR-ENAME**, so that the first name and the last name are separated by only one space. Use the statement, **CONCATENATE**.

Enter a width of 100 characters for the list (addition **LINE-SIZE** for the keyword, **REPORT**). Limit the length of the name to 30 characters and the length of the city to 20 characters. Separate the resulting field contents in your list by a vertical line (**SY-VLINE**).

Solution 4: Joining Infotypes

Task:

Create a report that documents the period in which the employee lived at which address under which name. The created list should contain the following information:

Personnel number

Start date

End date

Name

Place of residence

1. Process the records from the infotype, 0006, only which have the subtype 1.

Print data only for the periods for which address information is available.

Use a variable with the characteristics of the field, **PERNR-ENAME**, so that the first name and the last name are separated by only one space. Use the statement, **CONCATENATE**.

Continued on next page

Enter a width of 100 characters for the list (addition **LINE-SIZE** for the keyword, **REPORT**). Limit the length of the name to 30 characters and the length of the city to 20 characters. Separate the resulting field contents in your list by a vertical line (**SY-VLINE**).

a)



```

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-endda
            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,
         sy-vline,
         p0002-begda,
         sy-vline,
         p0002-endda,
         sy-vline,
         (30) name,
         sy-vline,
         (20) p0006-ort01.
ENDFORM.

```



Lesson Summary

You should now be able to:

- Understand join and intervals
- Join two or more infotypes in a single PROVIDE - ENDPROVIDE loop

Lesson: Field Projection

Lesson Overview

This lesson provides an overview of how to select, process, and generate logical data views on one or more infotype fields. You will also learn to combine the two logical views of infotype data, join and projection, and read time-dependent table entries.



Lesson Objectives

After completing this lesson, you will be able to:

- Process one or more infotypes
- Combine the logical views of infotype data - join and projection
- Read time-dependent table entries

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The personnel department needs a report for the data from the different infotype fields.

Projection and Contraction

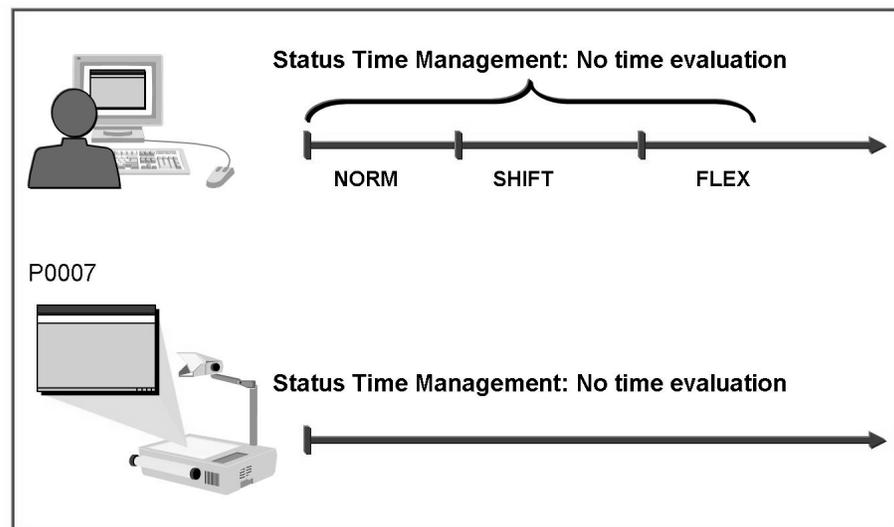


Figure 40: Projection and Contraction

When you select one or more infotype fields for processing, it is called **projection**. Using projection for specific fields enables you to indicate that only the contents of the fields and time-based changes to the contents are relevant to an evaluation.

Like joins, projections are logical data views 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 the fields that are not included in the projection are in an undefined condition, which means they are no longer relevant to the output. If the contents of a field included in the projection remain unchanged in several data records, the validity periods of the data records are combined.

The process to combine 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.

```

Figure 41: Projection/Source Code

The * in the PROVIDE statement, which stands for all 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 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 allowed only for the **time constraints 1 and 2**.

Join and Projection

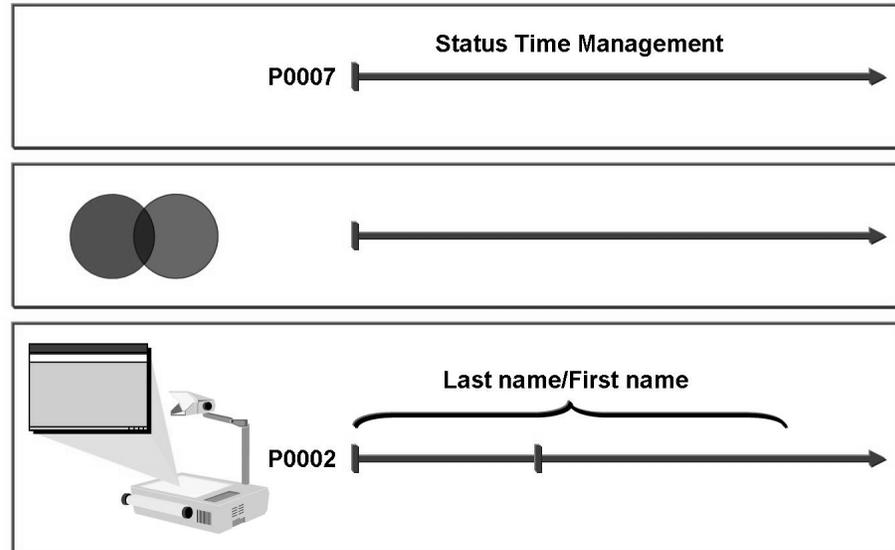


Figure 42: Join and Projection

These logical views of the infotype data, joins and projections, can be combined.

Data from several infotypes is read at the same time and new partial periods are created. Because fields are selected from infotypes only if they are relevant to an evaluation, the partial periods are combined. In this way, the validity periods with related contents are combined and the data is retrieved for an evaluation from the 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.
```

Figure 43: Join and Projection/Source Code

JOIN and PROJECTION can be combined in a PROVIDE statement.

Exercise 5: Projection to Infotype Fields

Exercise Objectives

After completing this exercise, you will be able to:

- Project to particular fields by joining two infotypes

Business Example

All the components of the HR system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The Board of Directors requires a list of employees with the values of the Status of Time Evaluation field in the Planned Working Time infotype.

Task:

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. Project to the fields, **P0002-NACHN** and **P0002-VORNA**, for infotype 0002. Project to the field **P0007-ZTERF** for the infotype, 0007.

Print the data for only the periods for which the address information is available.

The texts for the keys in the *Status of Time Recording* field are contained in the table, T555V. Before you read 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.

Enter a width of 100 characters for the list (addition **LINE-SIZE** for keyword **REPORT**).

Solution 5: Projection to Infotype Fields

Task:

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. Project to the fields, **P0002-NACHN** and **P0002-VORNA**, for infotype 0002. Project to the field **P0007-ZTERF** for the infotype, 0007.

Print the data for only the periods for which the address information is available.

The texts for the keys in the *Status of Time Recording* field are contained in the table, T555V. Before you read 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.

Enter a width of 100 characters for the list (addition **LINE-SIZE** for keyword **REPORT**).

a)



```
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'.
```

Continued on next page

```

CONCATENATE p0002-nachn p0002-vorna
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.

```



Lesson Summary

You should now be able to:

- Process one or more infotypes
- Combine the logical views of infotype data - join and projection
- Read time-dependent table entries

Lesson: Function Modules

Lesson Overview

This lesson deals with the various function modules available for Capital Management. The function modules enable various HCM functions such as the tasks to read infotypes without the logical database, determine the entry date of an employee, and update infotype records. It also explains how to use the different macros available for HCM.



Lesson Objectives

After completing this lesson, you will be able to:

- Use macros
- Process infotype records
- Read infotypes without the logical database
- Update infotype records

Business Example

All components of the HCM system have been implemented at your company, the Training International company. The company wants to use the special features of report programming within Human Resources. The Executive Board requires lists with valid data for an employee in a specified data selection period.

Overview of Macro and Function Modules



```

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.

```

Figure 44: Macro Modules

Like subroutines and function modules, macro modules can be used in HCM to modularize source code.

For documentation on key macros, see the information on *programming aids for the logical databases PNP and PAP* in SAP Library.

Similar to subroutines and function modules, you can use macro modules to modularize programs. Macro modules are frequently used in HCM.

The 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 the macros in reports that do not use the logical database, PNP, you must include the 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 Macros in the ABAP Programs table, TRMAC.

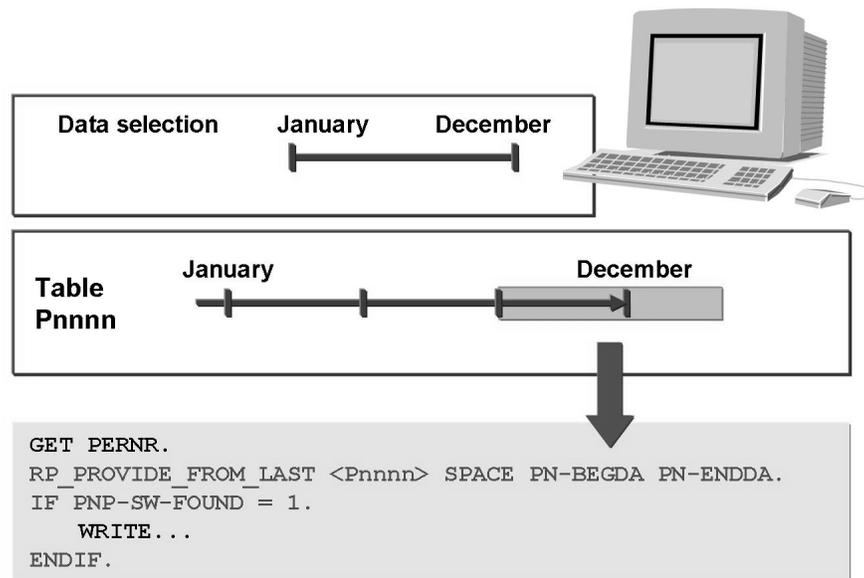


Figure 45: Processing a Specific Infotype Record

The `RP_PROVIDE_FROM_LAST` macro retrieves the last valid data record in the data selection period.

The parameters for the `RP_PROVIDE_FROM_LAST` are infotype, subtype, start date, and end date. If you do not want to specify a particular subtype, enter `SPACE`.

You can process not only the last valid data record in the data selection period, but also the first valid data record using the `RP_PROVIDE_FROM_FRST` macro.

The macro returns code. **PNP-SW-FOUND** has the value, 1, if a suitable entry exists in the infotype table for the specified period. If no entry is found, the value is 0.

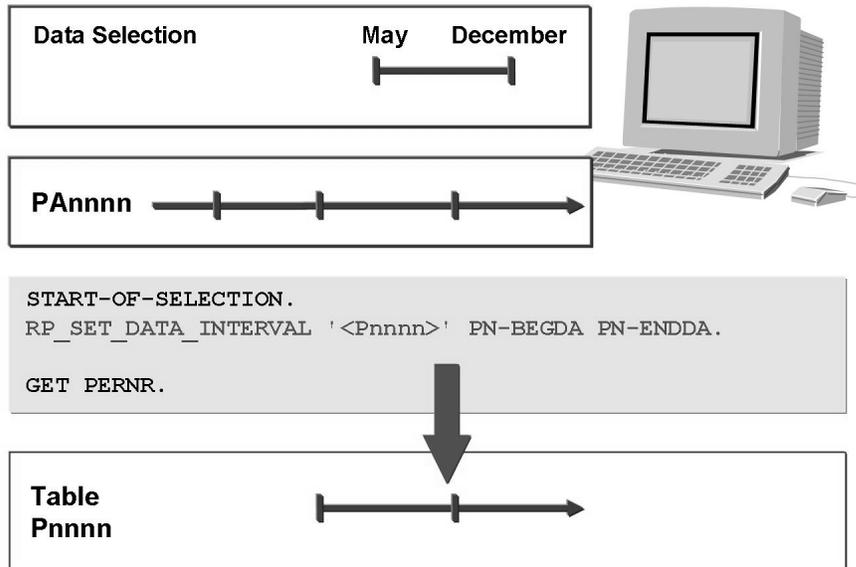


Figure 46: Importing Specific Infotype Records

When the START-OF-SELECTION event occurs, the RP_SET_DATA_INTERVAL macro ensures that the only records imported from the database table, PAnnnn, to the internal table, Pnnnn, are the ones that 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.
```

Arrows in the diagram point from the 'CABB' and '0001' values in the table to the corresponding values in the RP_PROVIDE_FROM_LAST macro call in the code block below.

Figure 47: Reading the Personnel Area/Subarea Table

You need to read the groupings stored in the Personnel Area/Subarea table because they are required as keys for other tables. This must take place individually for each personnel number to be processed. Each employee's assignment to a personnel area and subarea is stored in the infotype, 0001.

The parameters of the macro are:

- Personnel areas
- Personnel subareas

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_COUNTRY_GROUPING_GET**.



```

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

GET PERNR.
  RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGDA PN-ENDDA.
  . . . . .
  RP_READ_ALL_TIME_IT Y PN-BEGDA PN-ENDDA.
  . . . . .

  LOOP AT P2005.
    WRITE...
  ENDLOOP.

```

Figure 48: Importing Time Data

Because of the large amount of data in HCM, the infotypes, 2000 and 2999, should not be read when GET PERNR occurs. For this reason, the 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 later using the macro, **RP_READ_ALL_TIME_IT Y**, but only for the time interval specified by PN-BEGDA and PN-ENDDA.

Infotypes Without Logical Database

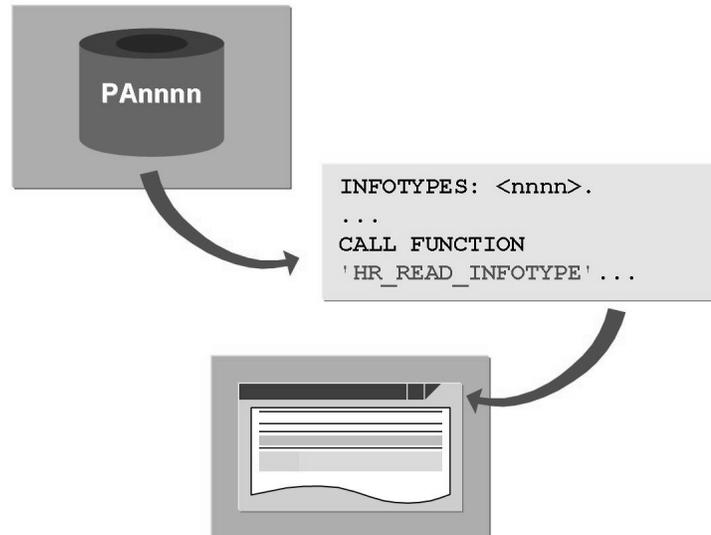


Figure 49: Reading Infotypes Without Logical Database (1)

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 HCM 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.

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`.

Notice that 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.

```

Figure 50: Reading Infotypes Without Logical Database (2)

The function module reads the HCM 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. An infotype record is selected if its validity period overlaps with the specified period.

The function module performs an authorization check. The following specifications are possible for the return code:

- 0: The return table contains all the required records.
- 4: The return table contains all the records but is incomplete due to missing authorization.
- 8: The return table is empty because no records were found with the specified criteria.
- 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 the reports, you can call the subroutine, **READ-INFOTYPE**, directly in the database program, **SAPDBPNP** (PERFORM READ-INFOTYPE(SAPDBPNP)). If this is the case, you declare the infotypes in the statement, INFOTYPES, with the supplement, **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.

```

Figure 51: Determining the Entry Date

You can use this function module to determine an employee's entry date. The following infotypes can be used to determine this date:

- P0000 Actions
- P0001 Organizational Assignment
- P0016 Contract Elements
- 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 the infotype, P0000.

If the infotype, P0016, is used, you can decide whether the fields, P0016-EINDT and 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 the infotype, P0041, the corresponding date type can be specified in the transfer parameter, DATUMSART. The date type, 01, is normally used for the technical entry date.

Updating Infotype Records



```

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.

```

Figure 52: Updating Infotype Records (1)

Before you change the employee data, you must lock the personnel number. You can do this with the function module, **BAPI_EMPLOYEE_ENQUEUE**.

Messages are returned in the parameter, **RETURN**. If an error occurs, this structure contains the following information:

- Message type (field *Type*)
- Message text (field *MESSAGE*)

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
          INFITY           = '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.

```

Figure 53: Updating Infotype Records (2)

You can use this function module to maintain the employee and applicant master data. You can specify one data record. All validations that would normally take place if the infotypes were maintained in the dialog mode with the individual maintenance screens are also carried out here. The module returns the same error messages as in the dialog mode. This means the error messages for the individual maintenance screens are displayed instead of being 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. Note: See the documentation for the module.

The following values are among the ones available for the parameter, **OPERATION**: MOD (change), COPY (copy), DEL (delete), INS (insert), LIS9 (delimit).

With the parameter, **DIALOG_MODE**, you specify whether the action is normally run in the background or it only runs in the background until an error occurs, or whether the changes are normally carried out in the dialog mode. Possible values:

- **0**: The changes are normally processed in the background. If an error occurs with an E or A message, the complete step is terminated and the module returns the corresponding error message in the structure, RETURN.
- **1**: The changes are normally processed in the background. If an error occurs, the system switches to the dialog mode so that the user can correct the entries.
- **2**: The changes are processed in the 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.
```

Figure 54: Updating Infotype Records (3)

If you cannot use the function module, HR_INFOTYPE_OPERATION, you can use the UPDATE statement. Notice that the system does not check the correctness of the new field contents when a direct DB update takes place.

In this example, the field, P0002-NATIO (nationality), in the DB table, PA0002, is updated directly. 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 the dialog mode.

The system field, SY-DBCNT, contains the number of changed records.

Authorization checks are not supported by the UPDATE statement and should be carried out at the program level.

Authorization Check for SQL Commands



```

CALL FUNCTION 'HR_CHECK_AUTHORITY_INFITY'
  EXPORTING
    TCLAS           = 'A'
    PERNR           = <person>
    INFITY          = '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.

```

Figure 55: Authorization Check for SQL Commands

Authorization checks are not supported by the SELECT statement, and for this reason they must be executed at the program level.

In this case, you must use the function module, **HR_CHECK_AUTHORITY_INFITY**, to check whether or not 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.

Exercise 6: Macros

Exercise Objectives

After completing this exercise, you will be able to:

- Use the `RP_SET_DATA_INTERVAL` macro to read the records only valid in the evaluation period
- Use the macro, `RP_PROVIDE_FROM_LAST`, to retrieve the last valid record in the data selection period
- Use the macro, `RP-READ-T001P`, to read the Personnel Area/Personnel Subarea table

Business Example

The Executive Board of the Training International Company requires a list of employees with recurring payments and deductions.

Task:

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. Use the `RP_SET_DATA_INTERVAL` macro to set that the records of only the infotype, 0014, are read for the evaluation period. Only the employees who have recurring payments or deductions in this period should appear on the list. If the employees do not have a record for the infotype, 0014, the internal table, P0014, is empty when the event, GET PERNR, occurs.

Continued on next page

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**.

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 a part of the key for the table, T512T (Wage Type Texts). Use the personnel area text in the field, **T001P-BTEXT**.

Read the corresponding wage type texts for the wage types for the infotype, 0014, from table, T512T. Before you read the 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.

Solution 6: Macros

Task:

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. Use the **RP_SET_DATA_INTERVAL** macro to set that the records of only the infotype, 0014, are read for the evaluation period. Only the employees who have recurring payments or deductions in this period should appear on the list. If the employees do not have a record for the infotype, 0014, the internal table, P0014, is empty when the event, GET PERNR, occurs.

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**.

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 a part of the key for the table, T512T (Wage Type Texts). Use the personnel area text in the field, **T001P-BTEXT**.

Read the corresponding wage type texts for the wage types for the infotype, 0014, from table, T512T. Before you read the 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.

a)

Continued on next page



```

REPORT zpsol040.
*-- Declaration
TABLES: pernr,
        t512t,           "Wage type text
        t001p.          "Plant section
INFOTYPES: 0001,       "Organ. Assignment
           0014.       "Recurr. Earn. & Deduc.

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

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

  LOOP AT p0014.
    CHECK p0014-betrg NE 0.
    PERFORM print USING
           p0014-lgart
           p0014-betrg
           p0014-begda
           p0014-endda.

  ENDLOOP.
*-----*
*      FORM PRINT      *
*-----*
*      Print List     *
*-----*
FORM print USING
  value(wtype)
  value(amount)
  value(begda)
  value(endda) .

```

Continued on next page

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

ENDFORM.                "print

*-----*
*          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.
```



Lesson Summary

You should now be able to:

- Use macros
- Process infotype records
- Read infotypes without the logical database
- Update infotype records



Unit Summary

You should now be able to:

- Understand join and intervals
- Join two or more infotypes in a single PROVIDE - ENDPROVIDE loop
- Process one or more infotypes
- Combine the logical views of infotype data - join and projection
- Read time-dependent table entries
- Use macros
- Process infotype records
- Read infotypes without the logical database
- Update infotype records

Internal Use SAP Partner Only

Internal Use SAP Partner Only



Test Your Knowledge

1. _____ is a logical database operation performed on the time axis.
Fill in the blanks to complete the sentence.
2. What does the time constraint 1 state?
Choose the correct answer(s).
 - A Saved without gaps and without overlaps
 - B Saved with gaps and without overlaps
 - C Saved without gaps and with overlaps
 - D Saved with gaps and with overlaps
3. The ___ in the PROVIDE statement is replaced by the five-digit field name of the infotype included in the projection.
Fill in the blanks to complete the sentence.
4. The data from several infotypes is read at the same time and new partial periods are created.
Determine whether this statement is true or false.
 - True
 - False
5. Which table record of the accessed infotype does the system normally read?
Choose the correct answer(s).
 - A The table record that is valid on the end date of the processed infotype.
 - B The table record that is valid on the start date of the processed infotype.
 - C The table record that is valid on the creation date of the processed infotype.
 - D The table record that is valid on the modification date of the processed infotype.

6. What do the first two letters in the macro name stand for?
Choose the correct answer(s).
- A Subroutine
 - B Function module
 - C Application
 - D Infotype
7. The transfer parameter STATUS2 is used to determine if infotypes P0000, P0001, P0016, and P0041 are taken into account.
Determine whether this statement is true or false.
- True
 - False
8. The macro return code PNP-SW-FOUND has the value 0 if a suitable entry exists in the infotype table for the specified period.
Determine whether this statement is true or false.
- True
 - False
9. Which parameter of the HR_INFOTYPE_OPERATION module has the MOD, COPY, DEL, INS, and LIS9 values available?
Choose the correct answer(s).
- A OPERATION
 - B DIALOG_MODE
 - C RETURN
 - D OPERATION_MODE



Answers

1. Join is a logical database operation performed on the time axis.

Answer: Join

2. What does the time constraint 1 state?

Answer: A

The time constraint 1 states saved without gaps and without overlaps.

3. The * in the PROVIDE statement is replaced by the five-digit field name of the infotype included in the projection.

Answer: *

4. The data from several infotypes is read at the same time and new partial periods are created.

Answer: True

The data from several infotypes is read at the same time and new partial periods are created. Because the fields are selected from infotypes only if they are relevant to an evaluation, the partial periods are combined.

5. Which table record of the accessed infotype does the system normally read?

Answer: A

The system normally reads the table record that is valid on the end date of the processed infotype.

6. What do the first two letters in the macro name stand for?

Answer: C

The first two letters in the macro name stand for application.

7. The transfer parameter STATUS2 is used to determine if infotypes P0000, P0001, P0016, and P0041 are taken into account.

Answer: False

The transfer parameter CHECK_INFOTYPES is used to determine if infotypes P0000, P0001, P0016, and P0041 are taken into account.

8. The macro return code PNP-SW-FOUND has the value 0 if a suitable entry exists in the infotype table for the specified period.

Answer: False

The macro return code PNP-SW-FOUND has the value 1 if a suitable entry exists in the infotype table for the specified period.

9. Which parameter of the HR_INFOTYPE_OPERATION module has the MOD, COPY, DEL, INS, and LIS9 values available?

Answer: A

The OPERATION parameter of the HR_INFOTYPE_OPERATION module has the MOD, COPY, DEL, INS, and LIS9 values available.

Unit 4

Repetitive Structures and List Display

Unit Overview

This unit shows how to process infotypes with repetitive structures and display data with the ABAP List Viewer. In practice, this is used to transfer report lists to applications such as Microsoft Word and Excel.



Unit Objectives

After completing this unit, you will be able to:

- Understand infotypes and repetitive structures
- Create a macro to evaluate repetitive structure
- List the functions of ABAP list viewer
- Explain the ALV data structure
- Create ALV field catalog
- Display the list using ABAP list viewer

Unit Contents

Lesson: Infotypes with Repetitive Structures.....	106
Lesson: List Display with ABAP List Viewer.....	111
Exercise 7: Date Specifications Infotype	117

Lesson: Infotypes with Repetitive Structures

Lesson Overview

This lesson helps you identify the occurrence of repetitive structures. In addition, this lesson helps you evaluate repetitive structures using a macro.



Lesson Objectives

After completing this lesson, you will be able to:

- Understand infotypes and repetitive structures
- Create a macro to evaluate repetitive structure

Business Example

All components of the HR system have been implemented at your company, the Training International company. The company wants to use the special features of report programming within Human Resources. The Executive Board needs an overview of the employees in a specific infotype in case of repetitive data.

Introduction to Repetitive Structures

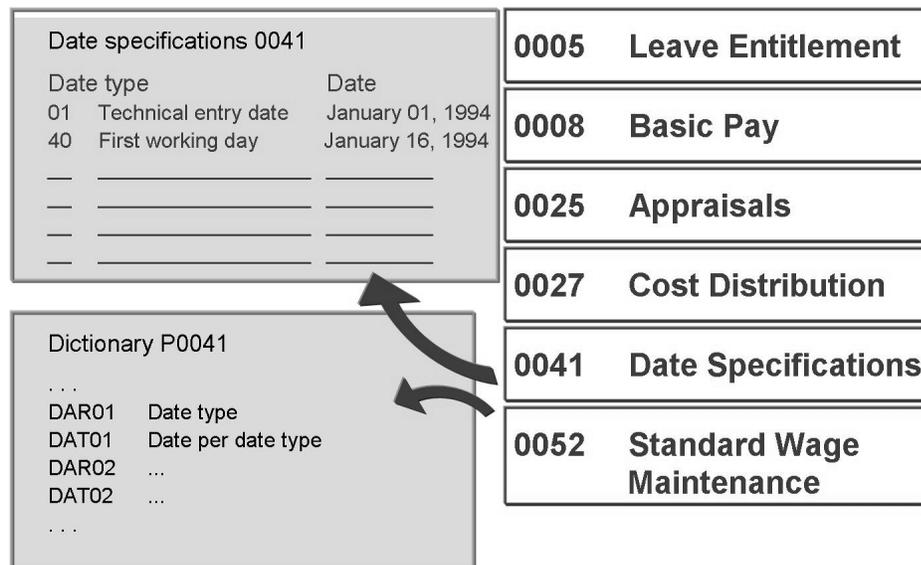


Figure 56: Infotypes with Repetitive Structures

On some infotype entry screens, data is entered in tables.

All of the fields in this table structure are named and defined in the infotype structure on which they are based.

In the Dictionary, repetitive structures can be recognized by the number at the end of the field name (Pnnnn-XYZnn).

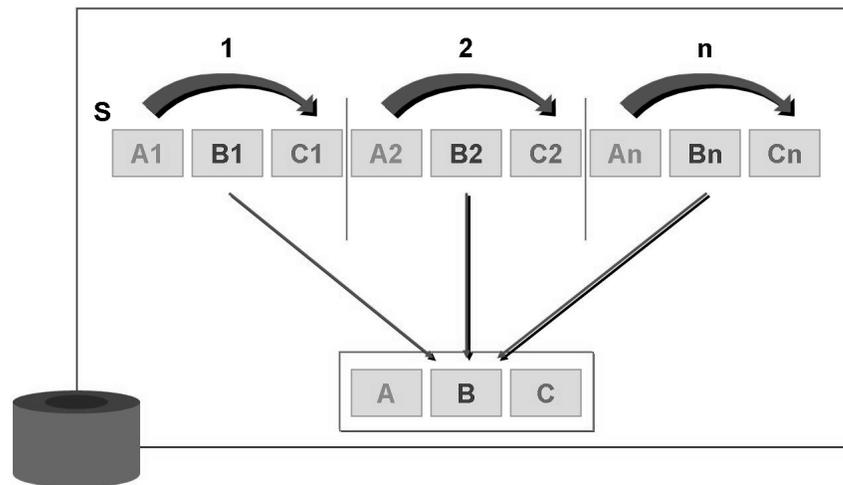


Figure 57: Repetitive Structures

If data is entered in a table on an infotype screen, it is stored on the database in a linear structure.

One line of the table is written to the database after another. The field names include the screen line number to ensure they are unique.

To evaluate such repetitive structures, you need to define starting point **S**, the increment value to the next restart point, the number of screen table lines (**n**), and a work area containing the field definition of a table line.

Evaluation of Repetitive Structures



```

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.

```

Figure 58: Repetitive Structures / Evaluation (1)

You can define the work area for the processing of repetitive structures as a field string with a structure that corresponds exactly to set of repeat fields in the relevant infotype table.

If some of the fields are not required for your evaluation, you can leave them out of a the defined structure. However, you cannot omit fields within the repetitive structure.



```

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.

```

Figure 59: Repetitive Structures / Evaluation (2)

This macro writes a record with a repetitive structure from the Date Specifications infotype to work area P0041.

The DO loop divides the repetitive structure into segments, and then places it into the defined work area on a block-by-block basis.

FROM <field name> is used to flag the starting point of the increment fields.

NEXT <field name> specifies the increment to the next group of repeat fields.

Alternative syntax:



```
WHILE... <condition>
VARY SPECIFICATION
FROM P0041-DAR01
NEXT P0041-DAR02.
ENDWHILE.
```

If unpacked data is contained in work area fields defined as packed, the evaluation is canceled. This can happen if the number of loop passes exceeds the number of repeat lines, or if the distance is not defined correctly.



Lesson Summary

You should now be able to:

- Understand infotypes and repetitive structures
- Create a macro to evaluate repetitive structure

Lesson: List Display with ABAP List Viewer

Lesson Overview

This lesson explains ABAP list viewer and its functions, such as the tasks to sort, print, and search data. In this lesson, you also learn about the data structure of ALV. Next, you learn how to create an ALV field catalog. Finally, the lesson explains how to display the list using the ALV field catalog.



Lesson Objectives

After completing this lesson, you will be able to:

- List the functions of ABAP list viewer
- Explain the ALV data structure
- Create ALV field catalog
- Display the list using ABAP list viewer

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The executive board needs data to be evaluated in a specific format.

Introduction to ALV

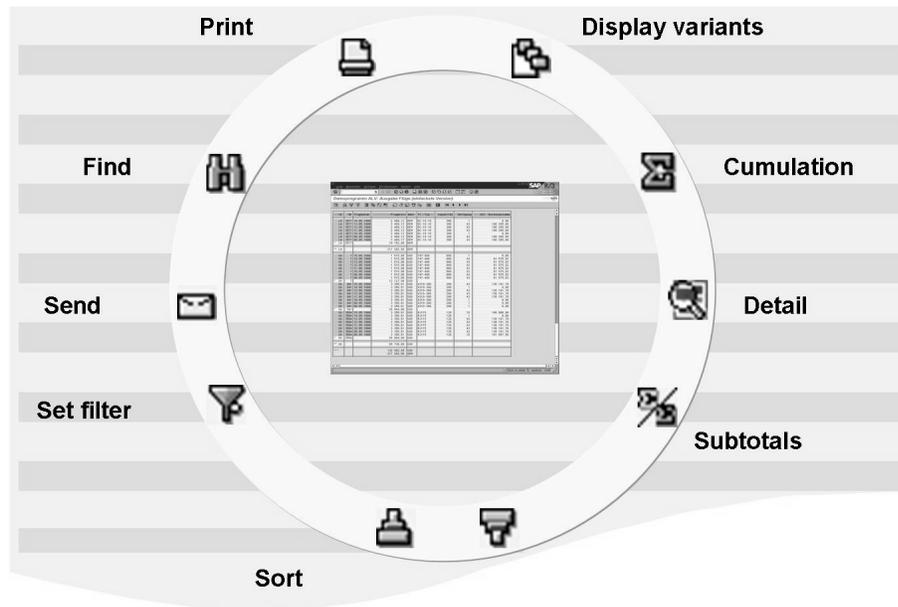


Figure 60: ABAP List Viewer: Functions

Standard reports (for example, RPLMIT00, RHSOLO00) use the ABAP List Viewer to format the data on display. This offers a number of available functions, for example:

- Sort data for one or more columns in ascending or descending order
- Print data
- Search for data
- Display data records in detail
- Set filter, in other words, reduce the displayed dataset (for example, one display data for one person)
- Hide and show fields on the list screen (using the display variant)
- Form totals
- Form subtotals (for example, total number of hours per week and per cost center)

These modifications can be made at runtime, however, they can also be stored as display variants.



```

. . .
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.

```

Figure 61: ALV Data Structures

The function module REUSE_ALV_GRID_DISPLAY enables you to use these functions in your own reports by providing

- an internal table with the superset of the information to be displayed
- a field catalog, and
- a structure with general layout information for the list display.

The data types required for the ABAP List Viewer are defined in the type pool **SLIS**. This type pool must therefore be included in the report with the statement **TYPE-POOLS**.

The field catalog transferred to the function module in the form of an internal table must have the type **slis_t_fieldcat_alv**.

You define the layout data for the list in the function module as a structure with type **slis_layout_alv**.

For the data to be displayed, you need an internal table with any structure. This table can contain more fields than are necessary for the list display. Only the fields specified in the field catalog and, if applicable, in the layout structure are used in the list display. The other fields in the internal table are ignored.

To store the display variants, you need a structure with the type **disvariant**.

ALV Variations



```
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.
```

Figure 62: ALV Field Catalog

The field catalog belonging to the output table is created in the program from which it is called. The creation of the field catalog and the explicit transfer can only be omitted if the structure of the internal table to be printed corresponds to a structure stored in the Dictionary, if all fields in this structure are printed on the list, and if the structure name is transferred to the function module using the parameter `I_STRUCTURE_NAME`.

- **fieldname:**
Name of field from internal output table that is described by the field catalog entry (mandatory parameter).
- **ref_tabname:**
Structure or table name of referenced field from Dictionary. This parameter is only filled if the internal output table field described by the current entry in the field catalog refers to the Dictionary (no program field).
- **ref_fieldname:**
Name of referenced field in Dictionary. This parameter is only filled if the internal output table field described by the current entry in the field catalog refers to the Dictionary (LIKE) and the field name in the internal output table differs from the name of the field in the Dictionary. If this field names are identical, it is sufficient to specify the Dictionary structure or table in the parameter `ref_tabname`.
- **key:**
Marks columns as a key column. Range of values: SPACE or 'X' displaying the key fields in color.

For information on additional parameters in the field catalog, see the documentation for the function module `REUSE_ALV_GRID_DISPLAY`.



```

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.

```

Figure 63: ALV Interface

Function module `REUSE_ALV_GRID_DISPLAY` is called at the **END-OF-SELECTION** event. Before this event, you must create the field catalog and, if applicable, define the layout specifications.

The structure `slis_layout_alv` contains parameters for display options, exceptions, totals, interaction, detail screens, colors, and so on. For descriptions of the parameters, see the documentation on the function module.

The parameter used in the example, `colwidth_optimize`, has the value range `SPACE` or `'X'`. `'X'` = optimizes column width so that the contents are displayed completely.

The parameters `I_SAVE` and `IS_VARIANT` enable you to store display variants. If you assign the value `'A'` to `I_SAVE`, the user can save a display variant as both a user-specific variant and a standard display variant. The user makes the selection when the dialog box for saving the display variant is shown.

If the structure transferred to the `IS_VARIANT` parameter is blank and the save parameter is still active (`I_SAVE` is not blank), the field `IS_VARIANT-REPORT = I_CALLBACK_PROGRAM` is set. For more information on both parameters, see the documentation on the function module.

Exercise 7: Date Specifications Infotype

Exercise Objectives

After completing 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

Business Example

All the components of the HR system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The personnel department requires a report that creates an overview of employees with records for the *Date Specifications* infotype. This overview can be used for further processing.

Task:

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 enable you to restrict the list to one or more date types. Only the employees with the records for the *Date Specifications* infotype, 0041, should appear in the overview.

The user can store display variants.

1. To place the last records for the 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.

Read the texts for the date types from the table, **T548T**.

Continued on next page

To determine the hiring date from the *Actions* infotype, use the function module, **RP_GET_HIRE_DATE**.

Set the layout parameters, **COLWIDTH_OPTIMIZE** and **ZEBRA**, to the value, "X".

You can use the following includes:

ZPSOL050_DATA_STRUCTURES: Contains the required data definitions.

ZPSOL050_FILL_FIELDCAT: Contains subroutines to fill the field catalog.

Solution 7: Date Specifications Infotype

Task:

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 enable you to restrict the list to one or more date types. Only the employees with the records for the *Date Specifications* infotype, 0041, should appear in the overview.

The user can store display variants.

1. To place the last records for the 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.

Read the texts for the date types from the table, **T548T**.

To determine the hiring date from the *Actions* infotype, use the function module, **RP_GET_HIRE_DATE**.

Set the layout parameters, **COLWIDTH_OPTIMIZE** and **ZEBRA**, to the value, "X".

You can use the following includes:

ZPSOL050_DATA_STRUCTURES: Contains the required data definitions.

ZPSOL050_FILL_FIELDCAT: Contains subroutines to fill the field catalog.

a)

Continued on next page



```

REPORT zpsol050.
*-- declaration

TABLES: pernr, t548t.           "Date Types
INFOTYPES: 0001,              "Org.Assignment
           0041.              "Date Specifications
TYPE-POOLS: slis.

DATA: BEGIN OF dtype,         "Workarea 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: stru_disvar TYPE disvariant.

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-endda.
rp_provide_from_last p0041 space pn-begda pn-endda.
IF pnp-sw-found EQ 0.
  REJECT.
ELSE.
  DO 12 TIMES VARYING dtype FROM p0041-dar01

```

Continued on next page

```

                                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_CALLBACK_PROGRAM          = 'ZPSOL060'
    I_GRID_TITLE                = 'Date Specifications'
    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.

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

*-----*
*      FORM READ_HIREDATE

```

Continued on next page

```

*-----*
*       Retrieving hiredate from infotype 0000
*-----*
FORM read_hiredate.
  CALL FUNCTION 'RP_GET_HIRE_DATE'
    EXPORTING
      PERSNR           = p0001-pernr
      CHECK_INFOTYPES = '0000'
    IMPORTING
      HIREFATE        = 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.

*-----*
*       FORM FIELDCAT_INIT
*-----*
FORM fieldcat_init USING p_fieldcat TYPE slis_t_fieldcat_alv.

```

Continued on next page

```
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.
```



Lesson Summary

You should now be able to:

- List the functions of ABAP list viewer
- Explain the ALV data structure
- Create ALV field catalog
- Display the list using ABAP list viewer



Unit Summary

You should now be able to:

- Understand infotypes and repetitive structures
- Create a macro to evaluate repetitive structure
- List the functions of ABAP list viewer
- Explain the ALV data structure
- Create ALV field catalog
- Display the list using ABAP list viewer

Internal Use SAP Partner Only

Internal Use SAP Partner Only



Test Your Knowledge

1. If data is entered in a table on an infotype screen, it is stored in the database in a linear structure.
Determine whether this statement is true or false.
 - True
 - False

2. If some of the fields are not required for your evaluation, you can omit the fields.
Determine whether this statement is true or false.
 - True
 - False

3. Which of the following tasks is not a function of ABAP list viewer?
Choose the correct answer(s).
 - A Searching for data
 - B Setting a filter
 - C Formatting totals
 - D Storing data

4. In ALV Data Structures, you define the layout data for the list in the function module as a structure with type, _____.
Fill in the blanks to complete the sentence.

5. The _____ parameter in the field catalog is used for the name of a referenced field in the dictionary.
Fill in the blanks to complete the sentence.

6. The function module, _____, is used to display data using ABAP list viewer.
Fill in the blanks to complete the sentence.



Answers

1. If data is entered in a table on an infotype screen, it is stored in the database in a linear structure.

Answer: True

If data is entered in a table on an infotype screen, it is stored in the database in a linear structure.

2. If some of the fields are not required for your evaluation, you can omit the fields.

Answer: False

If some of the fields are not required for your evaluation, you can leave them out of the defined structure. Notice that you cannot omit the fields within the repetitive structure.

3. Which of the following tasks is not a function of ABAP list viewer?

Answer: D

Storing data is not a function of ABAP list viewer.

4. In ALV Data Structures, you define the layout data for the list in the function module as a structure with type, slis_t_fieldcat_alv.

Answer: slis_t_fieldcat_alv

5. The ref_fieldname parameter in the field catalog is used for the name of a referenced field in the dictionary.

Answer: ref_fieldname

6. The function module, REUSE_ALV_GRID_DISPLAY, is used to display data using ABAP list viewer.

Answer: REUSE_ALV_GRID_DISPLAY

Unit 5

Cluster Database Tables in HCM

Unit Overview

This unit shows how to compile payroll results, where the data structures are defined, and how the data is stored in the database. It also covers how to modify the standard evaluation programs to suit the customer requirements and to create your own evaluation reports.



Unit Objectives

After completing this unit, you will be able to:

- Identify the features of PCLn DB tables
- Identify different types of PCLn DB tables
- Describe the structure of PCLn DB tables
- Import/export data using macros
- Import/export data using buffers
- Understand the data flow in payroll
- Create a function module to display payroll results using different parameters

Unit Contents

Lesson: Database Tables	130
Lesson: Importing Data	136
Exercise 8: Importing Cluster TX	141
Lesson: Payroll Results	147
Exercise 9: Displaying the Last Payroll Result	157
Exercise 10: Logical Database for Payroll Results	163

Lesson: Database Tables

Lesson Overview

This lesson helps you understand the features of PCLn database tables. It provides an overview of the two types of PCL databases, PCL1 and PCL2. In addition, it explains the structure of a PCLn database table.



Lesson Objectives

After completing this lesson, you will be able to:

- Identify the features of PCLn DB tables
- Identify different types of PCLn DB tables
- Describe the structure of PCLn DB tables

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The Personnel department wants to store data objects in a structured way.

Database Tables in HR



Figure 64: Database Tables in HR

The PAnnnn database tables contain all the HR data sorted by the infotype. They constitute the database for infotype entry screens and are evaluated by the HR logical database.

The PCLn database tables are a type of import/export database table. The PCL1, PCL2, and PCL3 database tables consist of either the database for subsequent programs, such as payroll runs or evaluations, or the database for subareas within Human Resources, such as travel expenses and recruitment.

Database Tables in PCLn

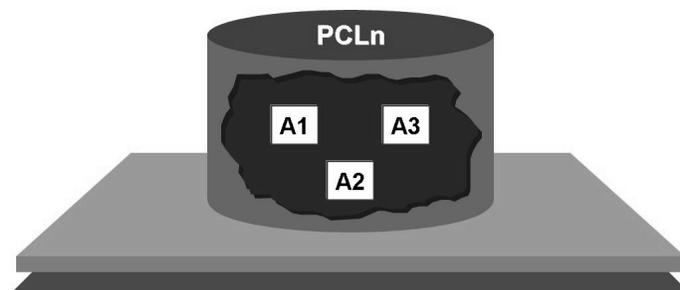


Figure 65: Database Tables PCLn

The database tables of the type, PCLn, are divided into subareas known as data clusters.

You identify data clusters 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 the information about the existing clusters and the database tables in which the clusters are stored.

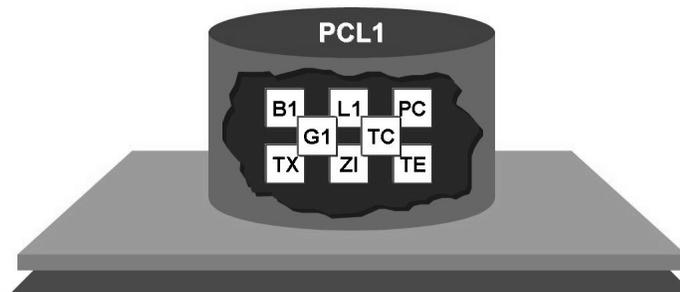


Figure 66: Database Table PCL1

Some of the data areas in the database table, PCL1, are:

- 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
- ZL: Interface PDC - Cost Accounting/Materials Management

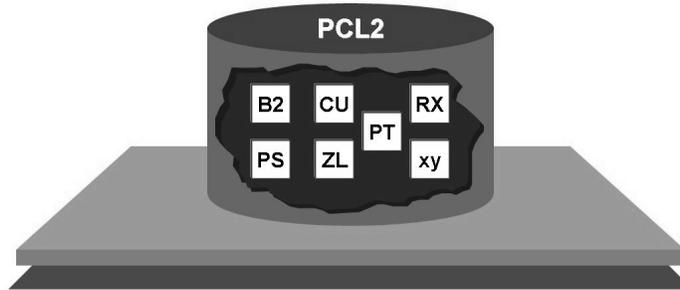


Figure 67: Database Table PCL2

The 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, where xy represents the relation ID. This either consists of Rn (n = HR country indicator), or the ISO code from table, T500L.
- ZL: Personal work schedule

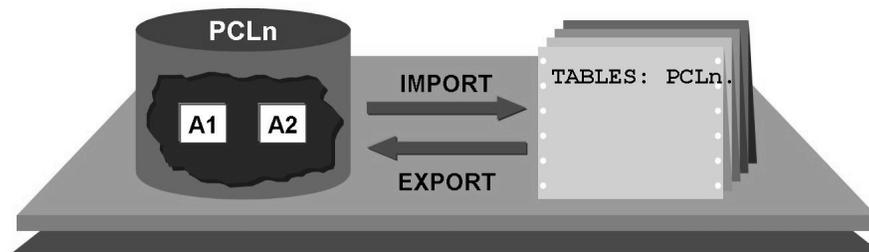


Figure 68: Database Table Administration/ PCLn

The PCLn import/export database tables are managed by the ABAP commands, IMPORT and EXPORT. You can use the 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
CLIENT	X	3	Client
RELID	X	2	Relation ID
SRTFD	X	40	Work area key
SRTF2	X	10	Sort field for duplicate key

<u>Example of International Payroll Results:</u>	
RELID	SRTFD
RX	00001911 00001
RX	00001911 00002
RX	00001911 00003
...	

Figure 69: Table Structure / PCLn

The structure of the PCLn database tables provides a basic structure for the individual data areas.

The name of each data area must include a two-character cluster name or a relation ID.

A key structure must also be defined. A size of 40 bytes is available in the field, SRTFD, for this purpose.

In the international payroll results, the field, **RELID**, contains the cluster name, RX, and the field, **SRTFD**, contains the eight-digit personnel number and 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 completes the **SRTF2** field.

The **CLIENT** field must only be completed if cluster data is to be imported from another client.



Lesson Summary

You should now be able to:

- Identify the features of PCLn DB tables
- Identify different types of PCLn DB tables
- Describe the structure of PCLn DB tables

Lesson: Importing Data

Lesson Overview

This lesson explains how to import and export data using macros and buffers.



Lesson Objectives

After completing this lesson, you will be able to:

- Import/export data using macros
- Import/export data using buffers

Business Example

All the components of the HR system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The personnel department needs to import and export data among database tables.

Introduction to Clusters



```
* 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, . . .
```

Figure 70: CLUSTER - Definition in Program, Rpcnxyz0

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 the convention, RPCnxyz0, where:

- n = 1 or 2 for PCL1 or PCL2
- xy = Cluster, such as, RX
- z = 0 for international clusters or country indicator from the 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, such as field strings and tables, are named individually for each cluster.

Importing Data with Macros



```

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.
. . .

```

Figure 71: Importing Data Using a Macro

To ensure consistency when data is exported and imported, the IMPORT/EXPORT commands are defined as macros.

You can 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**, to import infotype texts to the 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

Importing Data with Buffer

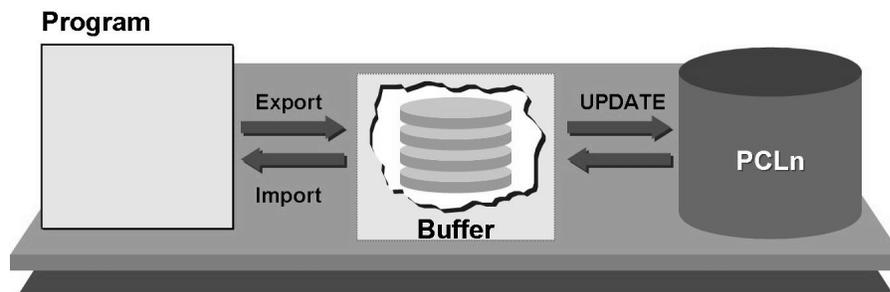


Figure 72: Importing/Exporting Using a Buffer

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. Notice that 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 results of a test run differ if the test runs are performed for several periods.

Using the buffer enables you to access the required results from the previous period.

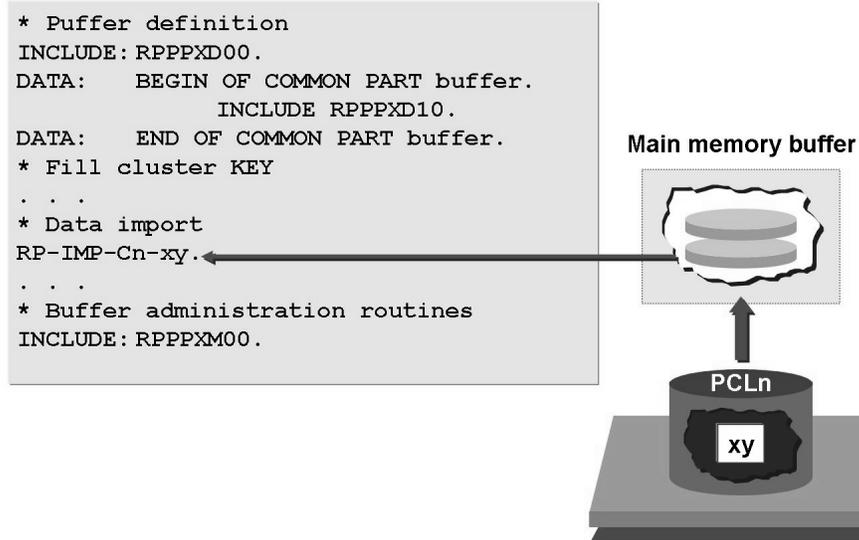


Figure 73: Importing Using a Buffer

If data is imported using macros, the data records are not read directly from the 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, RPPPXD00 and RPPPXD10. 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.

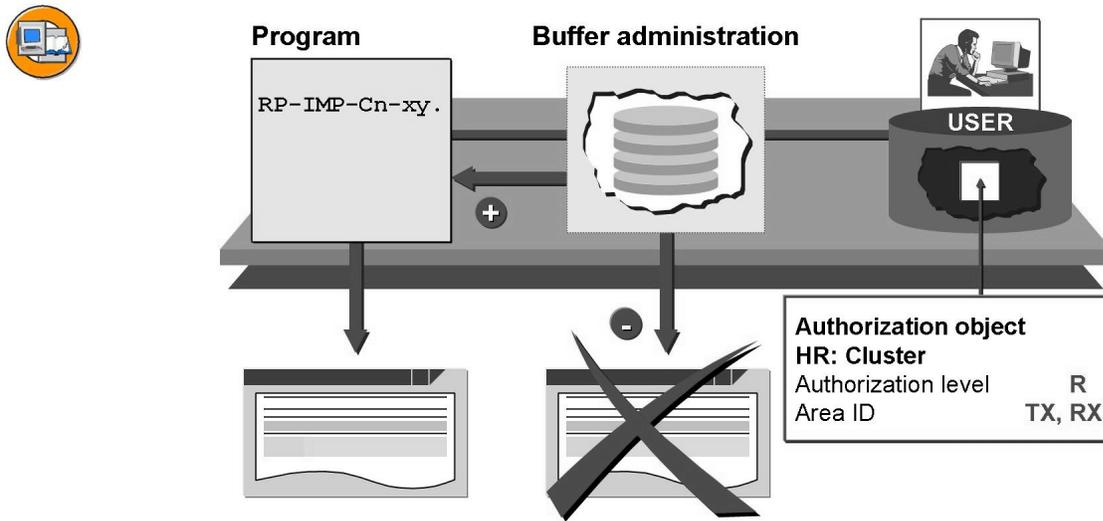


Figure 74: Cluster Authorization

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, and **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.

Exercise 8: Importing Cluster TX

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

All the components of the HR system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The departments at your company require a report that gives you an overview of the resources loaned to employees. This information is stored in the *Objects on Loan* infotype.

Task:

Write a report that lists only the employees who have received objects on loan. The report will also print the explanations that are entered by the administrator in the text fields.

1. The created 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

Take the name of the employee from the **PERNR-ENAME** field. The object on loan corresponds to the subtype for the infotype, 0040. To determine the subtype text, which is the description of the object on loan, read the table, **T591S**, with the keys, infotype and subtype. Display only the first 20 characters of the employee name and the subtype text.

The include program, **RPC1TX00**, contains the key for the 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 the table, P0040, with

Continued on next page

the **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 the cluster, TX. If the return code for the import is not zero, display an error message.

Solution 8: Importing Cluster TX

Task:

Write a report that lists only the employees who have received objects on loan. The report will also print the explanations that are entered by the administrator in the text fields.

- The created 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

Take the name of the employee from the **PERNR-ENAME** field. The object on loan corresponds to the subtype for the infotype, 0040. To determine the subtype text, which is the description of the object on loan, read the table, **T591S**, with the keys, infotype and subtype. Display only the first 20 characters of the employee name and the subtype text.

The include program, **RPC1TX00**, contains the key for the 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 the table, P0040, with the **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 the cluster, TX. If the return code for the import is not zero, display an error message.

a)



```
REPORT zpsol060.
TABLES: pernr,
        pcl1,
        pcl2,
        t591s.
```

"Subtype Table

Continued on next page

```

INFOTYPES: 0040.                                "Objects on Loan

INCLUDE: rpcltx00.                                "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.

```

Continued on next page

```

WRITE: / pernr-pernr,
      (20) p0001-ename,
          p0040-begda,
          p0040-endda,
          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 wurde geloescht.'(001).
  ENDIF.
ENDFORM.

```

```

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

```

```

*-- Subroutines
INCLUDE: rpppxm00.                                "Buffer-Subroutine

```



Lesson Summary

You should now be able to:

- Import/export data using macros
- Import/export data using buffers

Lesson: Payroll Results

Lesson Overview

This lesson helps you to understand the results and data flow in payroll. In addition, it describes how to create a function module to display payroll results using different parameters.



Lesson Objectives

After completing this lesson, you will be able to:

- Understand the data flow in payroll
- Create a function module to display payroll results using different parameters

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The personnel department needs payroll results that are given a specific criterion.

Overview of Data Flow

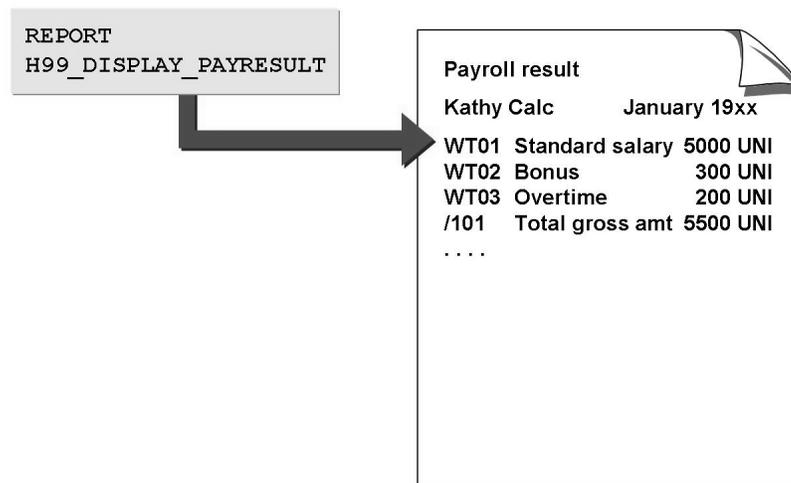


Figure 75: Payroll Results

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 to display personnel numbers

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

Each payroll result has a status indicator:

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

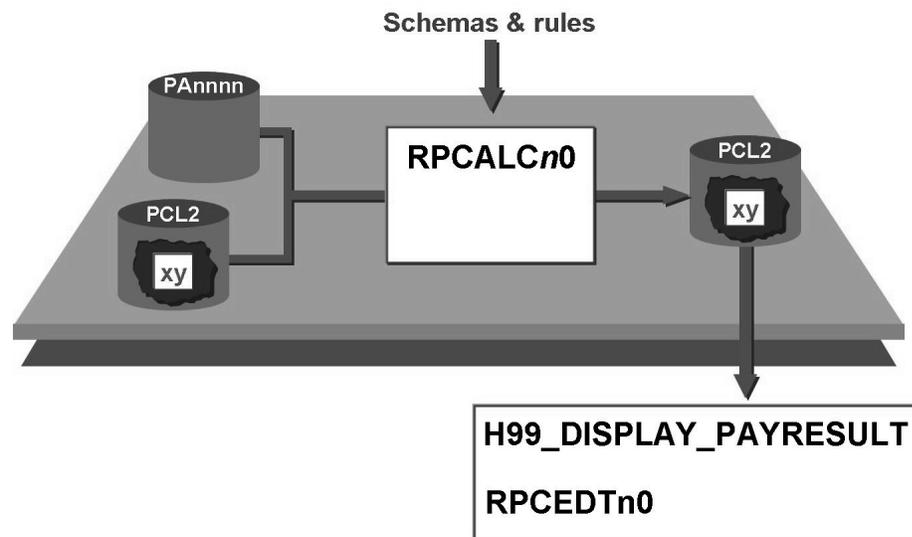


Figure 76: Data Flow in Payroll

The payroll driver, RPCALCn0, uses HR data, which is stored in the database tables, PAnnnn, and the last payroll result, which is 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, which contains the functions that call the subroutines contained in the payroll driver. In many cases, the function is enhanced by the rules for specific control of the subroutines.

The payroll result generated by the payroll driver is stored in the cluster, xy, of the database table, PCL2.

The report, H99_DISPLAY_PAYRESULT, displays the payroll results for PCL2 and, from Release 4.6C, replaces the 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.

For example, the report, RPCEDTn0, lists the formatted result as a payroll form (n = HR country indicator from table T500L).

Current Payroll Results by Cluster Directory



```
*Table containing directory of payroll results
DATA: directory TYPE TABLE OF PC261.

DATA: country LIKE T001P-MOLGA.
...
CALL FUNCTION 'CU_READ_RGDIR'
  EXPORTING
    PERSNR          = PERNR-PERNR
  IMPORTING
    MOLGA          = country
  TABLES
    IN_RGDIR       = directory
  EXCEPTIONS
    NO_RECORD_FOUND = 1
    OTHERS         = 2.
```

Figure 77: Reading the Cluster Directory

Table RGDIR contains the directory (cluster directory) for all the payroll results of an employee and is contained in the cluster, CU. A directory entry with the payroll area, for-period, in-period, status indicator, and the five-digit sequence number is required along with the personnel number, to construct the key for each payroll result for an employee.

The function module, CU_READ_RGDIR, reads the table, RGDIR, from the cluster, CU. The personnel number, the payroll directory of which 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.
. . .
CALL FUNCTION 'CD_READ_LAST'
  EXPORTING
    BEGIN_DATE      = PN-BEGDA
    END_DATE        = PN-ENDDA
  IMPORTING
    OUT_SEQNR       = number
  TABLES
    RGDIR           = directory
  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.

```

Figure 78: Determining Current Payroll Result (1)

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 the 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 and the table, RGDIR. The function module then gives you the sequential number, OUT_SEQNR, for the current result, A, 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.

```

Figure 79: Determining Current Payroll Result (2)

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 `PAYic_RESULT` (`ic` = ISO code).

You can use the `READ ONLY INTERNATIONAL` parameter to indicate 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.

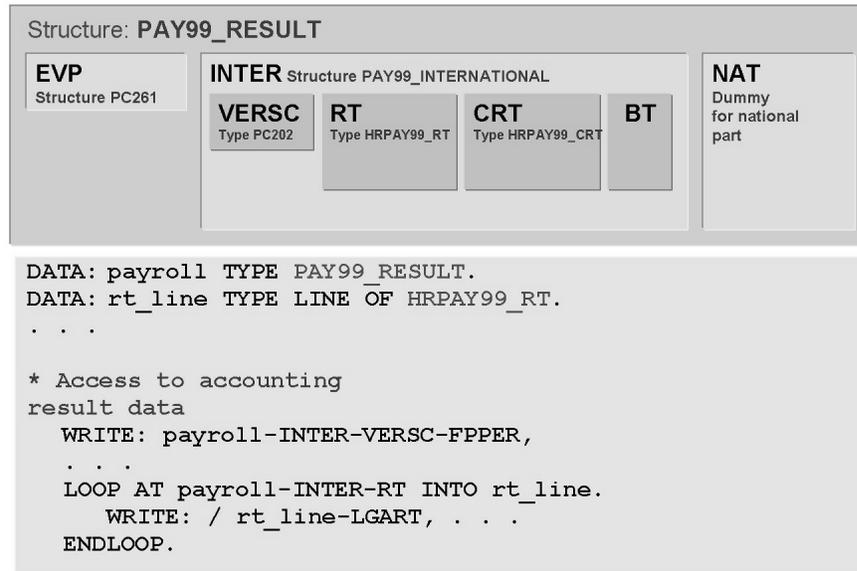


Figure 80: Data Structure for Payroll Results

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. For example, these components contain tables, such as RT and CRT, and structures, such as VERSC, for payroll results. In the cluster, RX, NAT consists of a dummy field.

The structures, PAY ic _RESULT (ic = ISO code, such as PAYUS_RESULT for the USA), exist for the country-specific results. In this case, 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).

Logical Database for Payroll Results

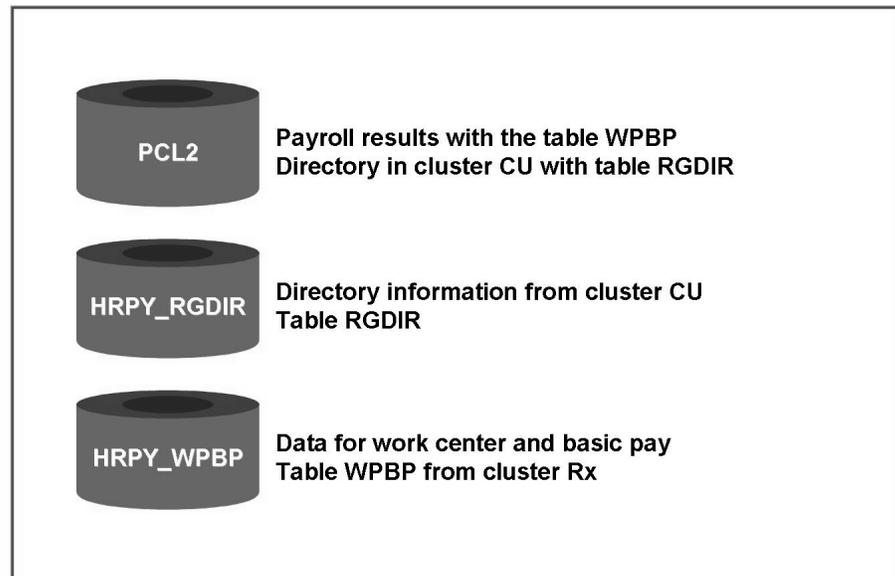


Figure 81: Logical Database for Payroll Results

Payroll results are saved in the cluster table, PCL2, to improve performance and free memory space. Notice that the logical database requires some of the 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.

Notice that this does not apply to the payroll records created in previous releases. To make this data available for evaluation, you must start the 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 the 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*, which is the 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.
...

```

Figure 82: Event GET PAYROLL

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 occurs only for the 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 the selected payroll results are processed. You can issue summaries when this event occurs.

You can use the *Result Status* to limit the selection of payroll results.

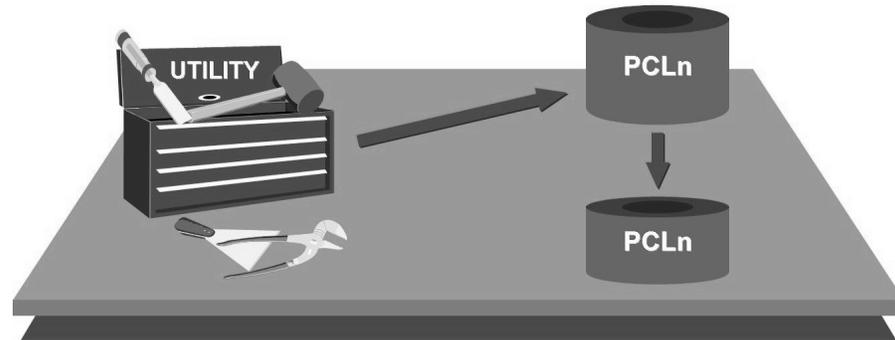


Figure 83: Utilities/Deleting a Cluster

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. In this context, n stands for the database tables, PCL1 or PCL2.

You can delete payroll results using the program, RPUDEL20.



Caution: The deletion utilities should only be used to test the archive data to reduce the amount of data in the database.

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Exercise 9: Displaying the Last Payroll Result

Exercise Objectives

After completing 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 results table `RT`

Business Example

The payroll department at the Training International company needs a list of wage types from the results table for the current (last) payroll result in a for-period.

Task:

Write a report to read and display the results of the international payroll to the 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. Take the name of the employee from the `PERNR-ENAME` field. The information about the for-period and the in-period can be found in the fields for the `VERSC` structure in the cluster, `RX`.

First, use the function module, `CU_READ_RGDIR`, to read the table, `RGDIR`, from the cluster, `CU`. To determine the sequential number for the last payroll result, transfer the table, `RGDIR`, to the function module, `CD_READ_LAST`.

Next, read the payroll result for the sequential number you have determined using the function module, `PYXX_READ_PAYROLL_RESULT`.

The wage types and amounts are saved in the `RT` table. The texts belonging to the wage types can be found in the table, `T512T`. Display only the first 15 characters of the employee name.

Solution 9: Displaying the Last Payroll Result

Task:

Write a report to read and display the results of the international payroll to the 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. Take the name of the employee from the **PERNR-ENAME** field. The information about the for-period and the in-period can be found in the fields for the **VERSC** structure in the cluster, RX.

First, use the function module, **CU_READ_RGDIR**, to read the table, RGDIR, from the cluster, CU. To determine the sequential number for the last payroll result, transfer the table, RGDIR, to the function module, **CD_READ_LAST**.

Next, read the payroll result for the sequential number you have determined using the function module, **PYXX_READ_PAYROLL_RESULT**.

The wage types and amounts are saved in the **RT** table. The texts belonging to the wage types can be found in the table, **T512T**. Display only the first 15 characters of the employee name.

a)

```
REPORT zpsol070.
TABLES: pernr,
        t512t.                                "Wage type texts
INFOTYPES: 0001.                             "Organizational Assignment
*Table data containing directory to PCL2 payroll results file
DATA: directory TYPE TABLE OF pc261.
DATA: payroll TYPE pay99_result.
DATA: rt_header TYPE LINE OF hrpay99_rt.

DATA: country LIKE t001p-molga,
      number  LIKE pc261-seqnr.
```

Continued on next page

```

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_rmdir        = 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
    CHANGING
      payroll_result = payroll
  EXCEPTIONS
    illegal_isocode_or_clusterid = 1
    error_generating_import      = 2

```

Continued on next page

```

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,

```

Continued on next page

```
        rt_line-betrg 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.
```

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Exercise 10: Logical Database for Payroll Results

Exercise Objectives

After completing this exercise, you will be able to:

- Process payroll results using the logical database for payroll results

Business Example

The payroll department at the Training International company needs a list of wage types from the results table for the current (last) payroll result in a for-period.

Task:

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

Solution 10: Logical Database for Payroll Results

Task:

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

a)

```
REPORT zpsol080.

TABLES: pernr, pyorgscreen, pytimescreen, t512t.

NODES: payroll TYPE pay99_result.

INFOTYPES: 0001.

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,
```

Continued on next page

```
        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.
```



Lesson Summary

You should now be able to:

- Understand the data flow in payroll
- Create a function module to display payroll results using different parameters



Unit Summary

You should now be able to:

- Identify the features of PCLn DB tables
- Identify different types of PCLn DB tables
- Describe the structure of PCLn DB tables
- Import/export data using macros
- Import/export data using buffers
- Understand the data flow in payroll
- Create a function module to display payroll results using different parameters

Internal Use SAP Partner Only

Internal Use SAP Partner Only



Test Your Knowledge

1. For each subarea within Human Resources, a separate _____ is defined.
Fill in the blanks to complete the sentence.
2. The data area, B2, in the PCL2 DB table stands for time events.
Determine whether this statement is true or false.
 - True
 - False
3. A key structure is defined in the _____ field.
Fill in the blanks to complete the sentence.
4. The naming convention for macros is RP-IMP-Cn-xy, where n and xy stand for:
Choose the correct answer(s).
 - A Filename and cluster name
 - B Number of employees and cluster name
 - C PCL DB table and country-specific cluster
 - D Cluster name and ISO code
5. If data is read using a buffer, the system ignores the cluster authorization.
Determine whether this statement is true or false.
 - True
 - False
6. The processing logic is imported by the program in the form of a _____.
Fill in the blanks to complete the sentence.
7. Explain the function module, CU_READ_RGDIR.



Answers

1. For each subarea within Human Resources, a separate data key is defined.

Answer: data key

2. The data area, B2, in the PCL2 DB table stands for time events.

Answer: False

The data area, B2, in the PCL2 DB table stands for time accounting results.

3. A key structure is defined in the SRTFD field.

Answer: SRTFD

4. The naming convention for macros is RP-IMP-Cn-xy, where n and xy stand for:

Answer: A

The naming convention for the macros is RP-IMP-Cn-xy, where n and xy stand for filename and cluster name, respectively.

5. If data is read using a buffer, the system ignores the cluster authorization.

Answer: False

If data is read using a buffer, the system checks the cluster authorization.

6. The processing logic is imported by the program in the form of a schema.

Answer: schema

7. Explain the function module, CU_READ_RGDIR.

Answer: This function module reads the table, RGDIR, from the cluster, CU. If the MOLGA parameter is active, the function module returns the HR country indicator.

Unit 6

Data Structures in Personnel Planning

Unit Overview

This unit provides an overview of the data model and the relationship among objects. It also describes the Personnel Planning data structures.



Unit Objectives

After completing this unit, you will be able to:

- Explain the data model
- Identify the relationships between objects
- Explain infotypes and objects/infotype status
- Describe the infotypes structures in Personnel Planning
- Explain logical structures and database tables

Unit Contents

Lesson: Functions and Data Structure	172
Lesson: Personnel Planning Data Structures	180
Exercise 11: Data Structures in Personnel Planning	191

Lesson: Functions and Data Structure

Lesson Overview

This lesson helps you understand the data model. You also learn about the relationships between objects. Finally, you learn about infotypes and the different object/infotype status.



Lesson Objectives

After completing this lesson, you will be able to:

- Explain the data model
- Identify the relationships between objects
- Explain infotypes and objects/infotype status

Business Example

All the components of the HR system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The executive board needs to create a data model based on the relationships between the various objects in the organization.

Overview of Data Model

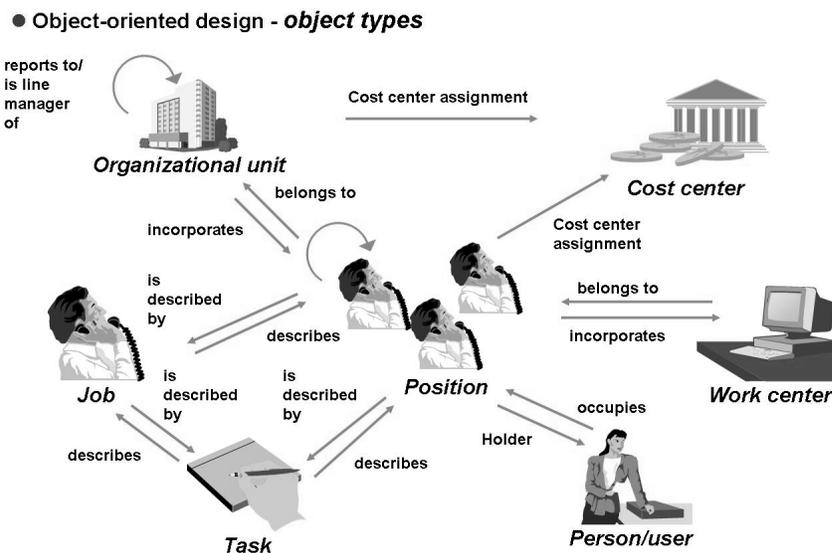


Figure 84: Personnel Planning Data Model (1)

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 shown as an independent object with individual characteristics. The objects are created and maintained separately. Relationships are used to link one object to the other (see the figure). This creates a network that enables 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. To do this, you define new object types and establish 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, which consists of object types and relationships, also constitutes the basis of other applications within Personnel Planning, such as Training and Event Management, including business event hierarchies and Personnel Development. For example, a qualification catalog is a part of this model.



● Object-oriented design - *relationships*

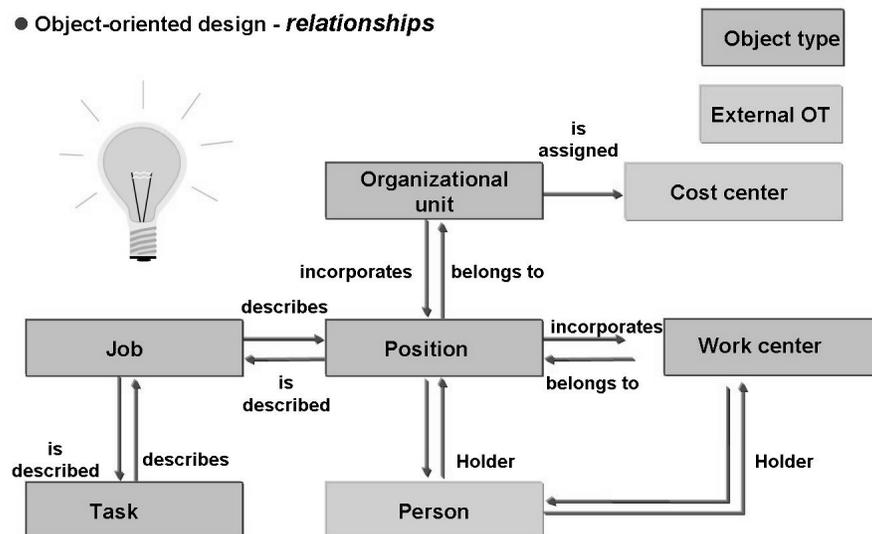


Figure 85: Personnel Planning Data Model (2)

You can use **relationships** to define the links between individual object types in the system.

Relationships are normally defined in two directions, $A = \textit{bottom up}$ and $B = \textit{top down}$.

For example, the relationship, "003", ("incorporates/belongs to") is defined in the following directions:

- $A003$ "belongs to" and $B003$ "incorporates"

Notice that 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 the 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, which means **external object types without the inverse relationship**.

Relationship between Objects



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

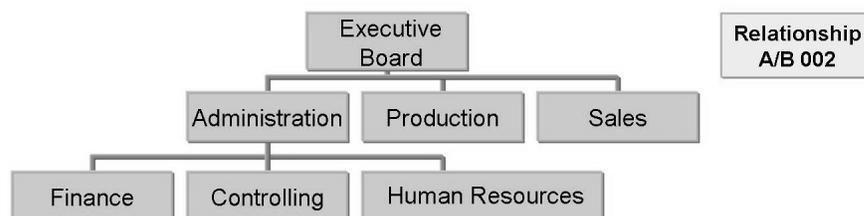


Figure 86: Relationship Between Organizational Units

Relationships are normally stored on the database in two directions.



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

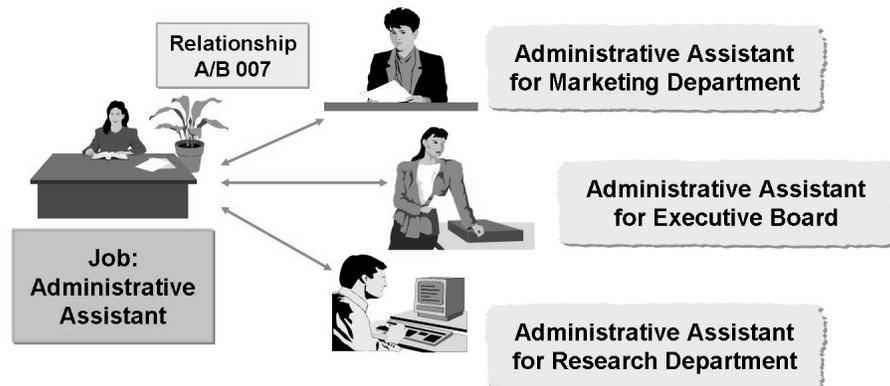


Figure 87: Relationship Between Jobs and Positions

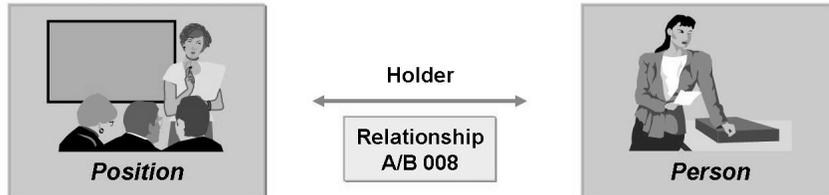
The term, "job", is used as a general classification, such as "secretary" or "head of department".

More than one employee can have the same job. For example, 20 employees can have the job of a secretary.

A position signifies the individual assignment of an employee in the company, such as a sales manager or a 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.

Figure 88: Relationship Between Person and Position

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

Infotypes



Object characteristics are maintained in the form of infotypes.

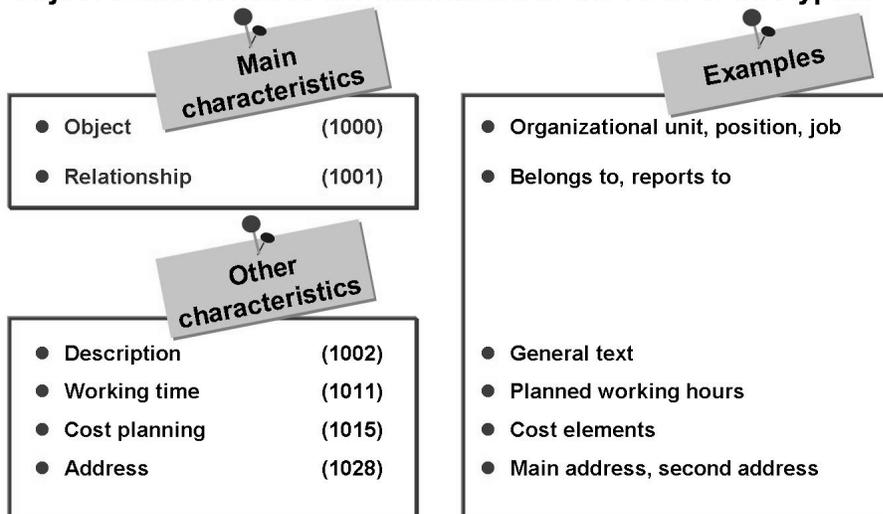


Figure 89: Infotypes

The *Object* infotype performs key functions. It enables you to create new objects, such as organizational units and jobs. In addition, it determines the validity period of all the other infotypes that describe the characteristics of the existing objects.

The **Object** infotype (1000) defines the existence of an object. 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)

After you have used this infotype to create an object, you can use the other infotypes such as *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 the company.

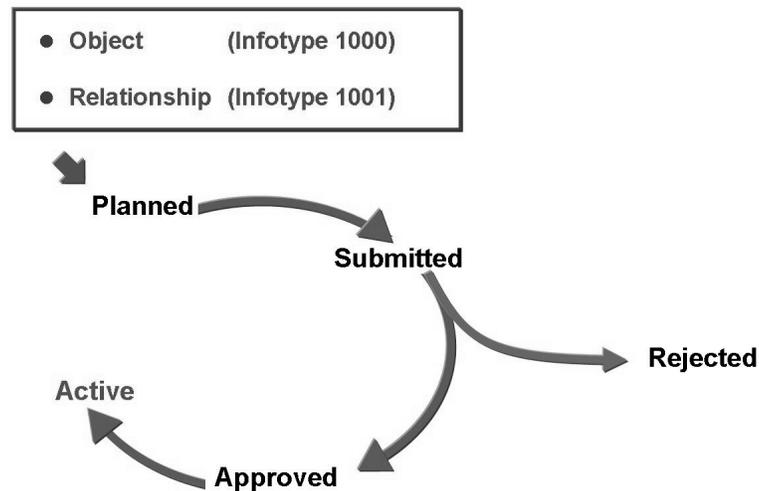


Figure 90: Object/Infotype Status

One of the following status is assigned to each object/infotype:

Active: Means that an object or infotype record can be currently 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 a 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*.



Lesson Summary

You should now be able to:

- Explain the data model
- Identify the relationships between objects
- Explain infotypes and objects/infotype status

Lesson: Personnel Planning Data Structures

Lesson Overview

This lesson describes the infotype structures in Personnel Planning. It also explains logical structures and database tables.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the infotypes structures in Personnel Planning
- Explain logical structures and database tables

Business Example

All the components of the HR system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. You want to draw up your own evaluations of Personnel Administration data. To do this, you must use the data structures in Personnel Planning.

Personnel Planning Data Structures



- Transparent tables *Dictionary field definition*

Example: HRP1003

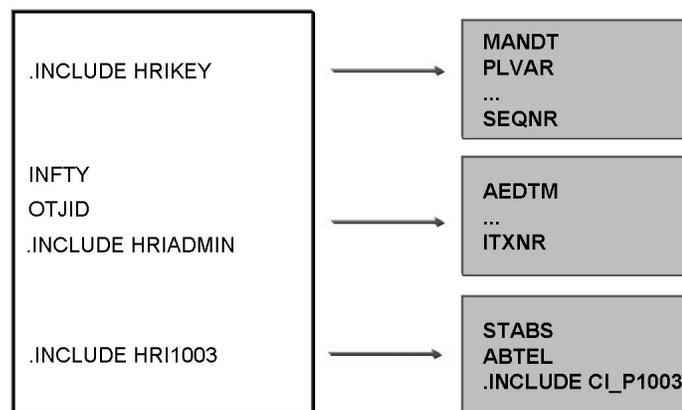


Figure 91: Personnel Planning Transparent Tables

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 the administrative information about the infotype record, which includes among 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 hold the actual information and data about the infotype.

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.

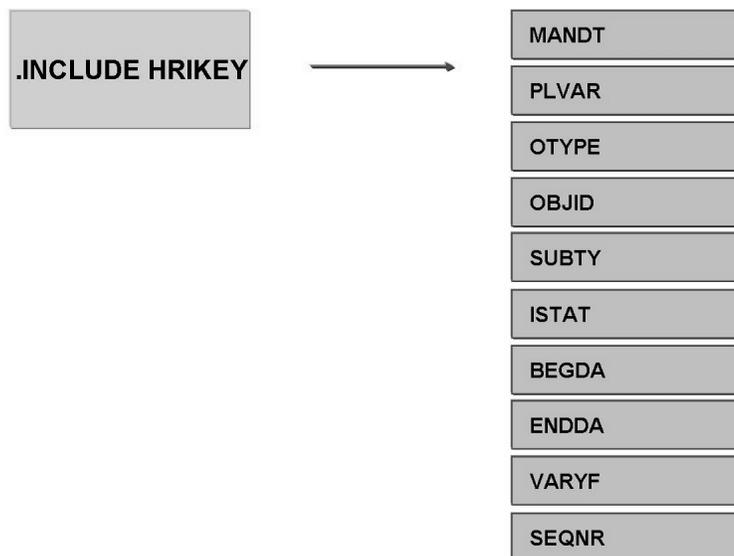


Figure 92: Key Fields Infotype Personnel Planning

The primary key of the transparent table, HRPnnnn, is made up of the following key fields:

- **MANDT**: Signifies a participant in the system that is legally and organizationally self-contained.
- **PLVAR**: Signifies a one- or two-digit alphanumeric **plan version key** that distinguishes different organizational plans.
- **OTYPE**: Represents a one- or two-digit alphanumeric **object type key**.
- **OBJID**: Represents an eight-digit numerical, nonmnemonic key that displays an individual object. For example, an organizational unit, a qualification, or an event.
- **SUBTY**: Indicates subtypes, which are subdivisions of infotypes. The subtypes of an infotype can have different time constraints and create their own history.
- **ISTAT**: Signifies a 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**: Indicates the valid from date.
- **ENDDA**: Indicates the valid to date.
- **VARYF**: Represents the variation field, which contains the target object for relationships.
- **SEQNR**: Indicates the **sequential number** that allows you to differentiate infotype records with 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.

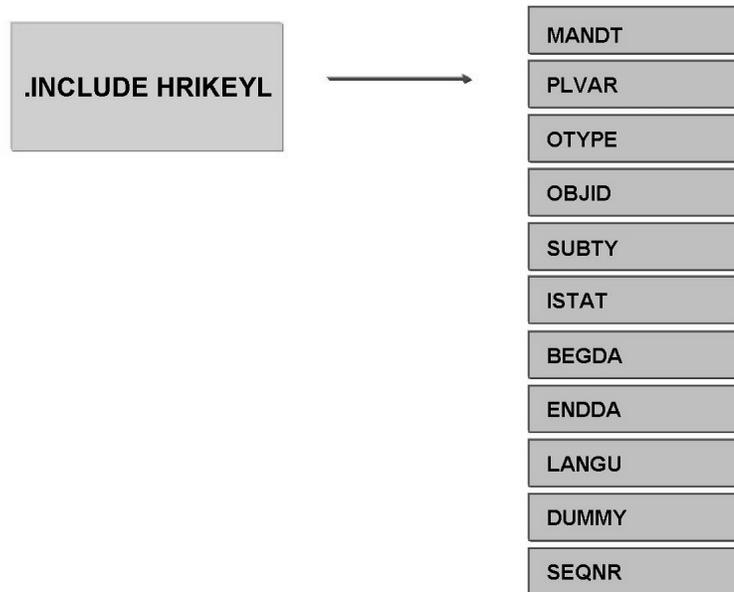


Figure 93: Language-Specific Infotypes Personnel Planning

Language-specific infotypes 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 the 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, as compared with the nonlanguage-specific infotype, *HRIKEYL*. The language of the infotype record is stored in the first position of the VARYF field. This means that 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 the other key fields are identical.

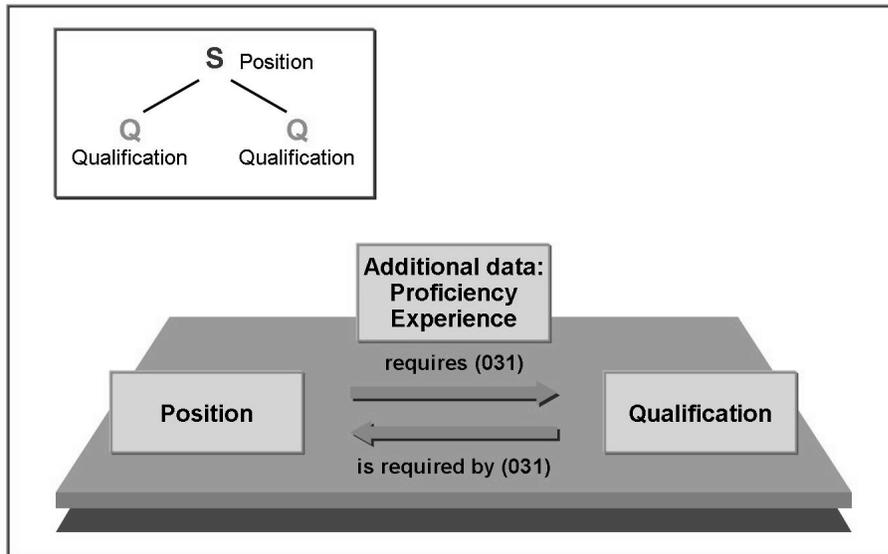


Figure 94: Additional Data on Relationships (1)

Additional data on relationships consists of attributes that cannot be stored in isolation as infotypes for the objects in question. This is because they indicate exclusively to the concrete relationship between two objects.

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 about 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**

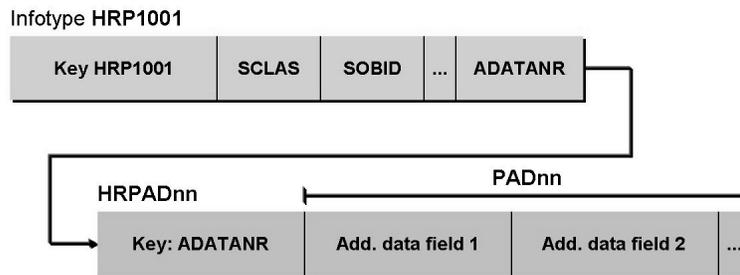


Table T77AD: Name of database table for an additional data structure

Figure 95: Additional Data on Relationships (2)

Because 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*.

The 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 the 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 *the 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 the table, *T77AR*.



● **Technical implementation - table infotypes**

Table infotype HRPnnnn

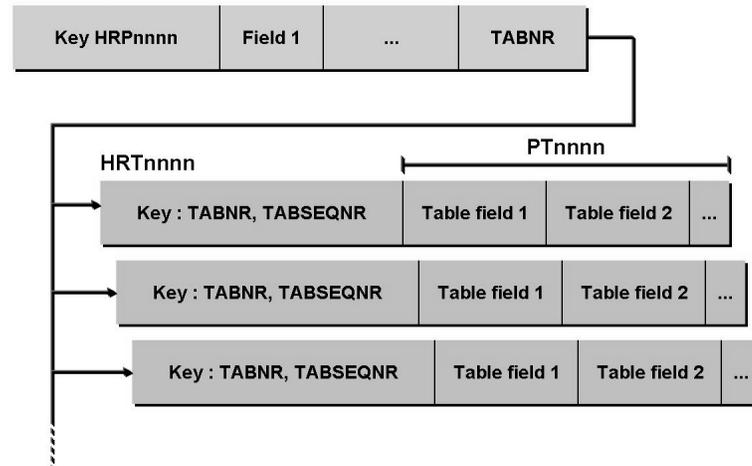


Figure 96: Table Infotypes Personnel Planning

Table infotypes signify the infotypes the data of which has a repetitive structure of any length, such as text with any number of lines and 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, the table, *HRPnnnn*, and the table part, the 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.

Similar to 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 *the 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
...	

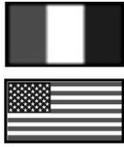


Figure 97: Country-Specific Infotypes Personnel Planning

The *NAT_INFITY* ("Country-specific infotype") field is used to label a country-specific infotype in the table, *T777D*.

The country-specific infotype is assigned to one or more country ids from the 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 also appear 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.

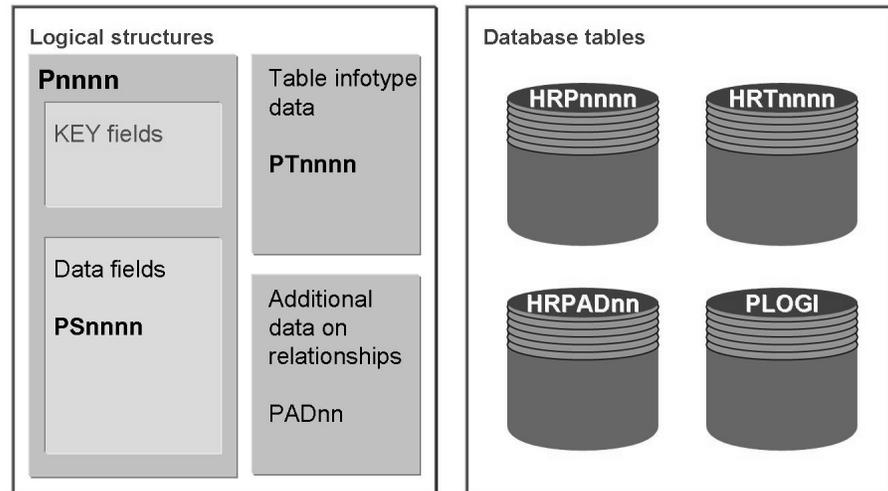


Figure 98: Logical Structures and Database Tables

The **Pnnnn** logical structure contains the key fields and data fields of an infotype. During interface definition, it is used in Personnel Administration among 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 the objects in the Personnel Planning database. You can use the **RHPLOGI0** report for display/set up. This object directory forms the basis of all the evaluations using the **PCH** logical database.

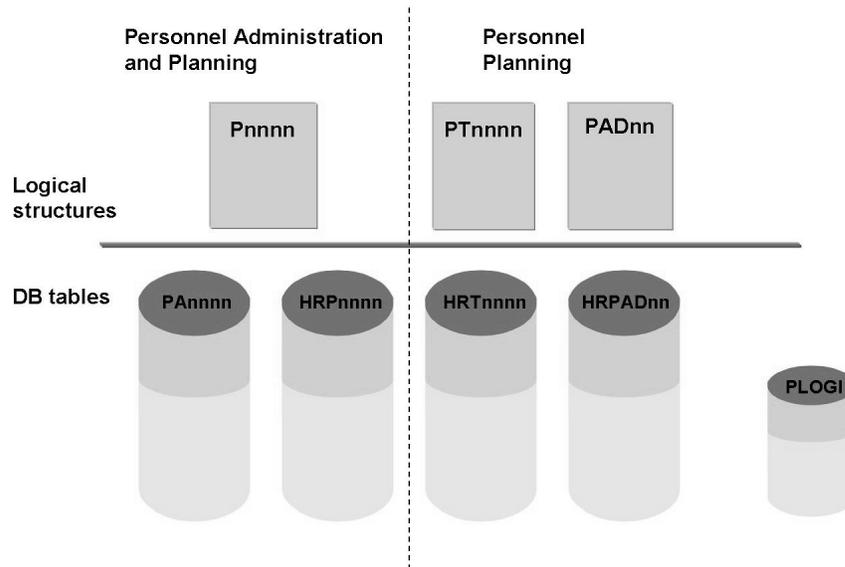


Figure 99: Overview of Logical Structures and Database Tables

The following elements exist for each nnnn infotype:

- A *Pnnn* logical structure.
- *APAnnnn* 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 the 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 for this reason, 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
HRIKEY(L)	Structure	-	No
HRInnnn	Structure	CI_Pnnnn	HRI0nnn-HRI8nnn: Only CI_Pnnnn HRI9nnn: Yes
Pnnnn	Structure	HRInnnn	P0nnn-P8nnn: No Only HRI9nnn
HRPnnnn	Transparent table	HRIKEY or HRIKEYL	HRP0nnn-HRP8nnn: No Only HRI9nnn
PTnnnn	Structure	-	PT0nnn-PT8nnn: No PT9nnn: Yes
HRTnnnn	Transparent table	PTnnnn	HRT0nnn-HRT8nnn: No Only PT9nnn
CI_Pnnnn	Structure	-	Yes

Figure 100: Overview of PD Data Structures

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

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Exercise 11: Data Structures in Personnel Planning

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

As a member of the IT team, the personnel department wants you 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.

Task 1:

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

1. Name the concepts that form the basis of this design.
2. How can you extend the Personnel Planning data model?

Task 2:

1. What is meant by additional data on relationships?
2. In which logical structure is additional data stored?

Task 3:

1. What is meant by a table infotype?
2. Which fields does the PTnnnn structure contain?

Solution 11: Data Structures in Personnel Planning

Task 1:

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

1. Name the concepts that form the basis of this design.
 - a) The following concepts form the basis of this design:
 - Object types
 - Relationships
 - Infotypes
2. How can you extend the Personnel Planning data model?
 - a) You can extend the Personnel Planning data model by defining new object types and new relationships in Customizing

Task 2:

1. What is meant by additional data on relationships?
 - a) Additional data on relationships is made up of attributes that cannot be stored in isolation for the objects in question because they indicate exclusively to the concrete relationship between two objects.
2. In which logical structure is additional data stored?
 - a) The additional data is stored in the P1001 logical structure in the ADATA field.

Task 3:

1. What is meant by a table infotype?
 - a) A table infotype is an infotype in which the data part has a repetitive structure.
2. Which fields does the PTnnnn structure contain?
 - a) The PTnnnn structure contains the repetitive data fields for table infotypes.



Lesson Summary

You should now be able to:

- Describe the infotypes structures in Personnel Planning
- Explain logical structures and database tables



Unit Summary

You should now be able to:

- Explain the data model
- Identify the relationships between objects
- Explain infotypes and objects/infotype status
- Describe the infotypes structures in Personnel Planning
- Explain logical structures and database tables



Test Your Knowledge

1. The cost center is an internal object type.
Determine whether this statement is true or false.
 - True
 - False

2. How many letters does a standard object type contain?
Choose the correct answer(s).
 - A Four
 - B Two
 - C Three
 - D Five

3. A _____ signifies the individual assignment of an employee in the company.
Fill in the blanks to complete the sentence.

4. You can create objects with the status, planned or _____.
Fill in the blanks to complete the sentence.

5. Which of the following key fields is a participant in the system that is legally and organizationally self-contained?
Choose the correct answer(s).
 - A PLVAR
 - B OBJID
 - C SUBTY
 - D MANDT

6. The PLOGI transparent table contains an index of all the objects in the Personnel Planning database.
Determine whether this statement is true or false.
 - True
 - False



Answers

1. The cost center is an internal object type.

Answer: False

The cost center is an external object type because it is not maintained in Organizational Management.

2. How many letters does a standard object type contain?

Answer: B

Each standard object type contains two letters.

3. A position signifies the individual assignment of an employee in the company.

Answer: position

4. You can create objects with the status, planned or active.

Answer: active

5. Which of the following key fields is a participant in the system that is legally and organizationally self-contained?

Answer: D

MANDT is a participant in the system that is legally and organizationally self-contained.

6. The PLOGI transparent table contains an index of all the objects in the Personnel Planning database.

Answer: True

The PLOGI transparent table contains an index of all the objects in the Personnel Planning database.

Unit 7

Logical Database PCH

Unit Overview

This unit covers formatting structures and evaluation paths. It also examines how to evaluate infotypes. In addition, this unit explains structural and sequential evaluations.



Unit Objectives

After completing this unit, you will be able to:

- Identify structure parameters
- Set structure conditions
- Maintain evaluation paths
- Create reports for sequential evaluations
- Create reports for structural evaluations
- Evaluate table infotypes

Unit Contents

Lesson: Structure Formats	198
Lesson: Evaluations-Source Code	206
Exercise 12: Creating a List of Staff Positions	215

Lesson: Structure Formats

Lesson Overview

This lesson helps you identify various structure parameters and set structure conditions. In addition, the lesson discusses how to maintain evaluation paths.



Lesson Objectives

After completing this lesson, you will be able to:

- Identify structure parameters
- Set structure conditions
- Maintain evaluation paths

Business Example

All components of the HCM system have been implemented at your company, the Training International company. The company wants to use the special features of report programming within Human Resources. The 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. For this, you need to have an understanding of the evaluation paths.

Types of Evaluations

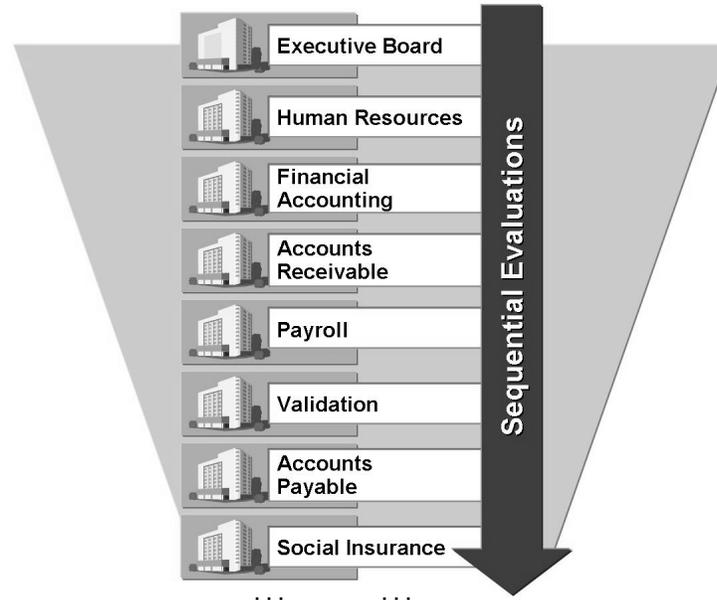


Figure 101: Sequential Evaluations

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



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

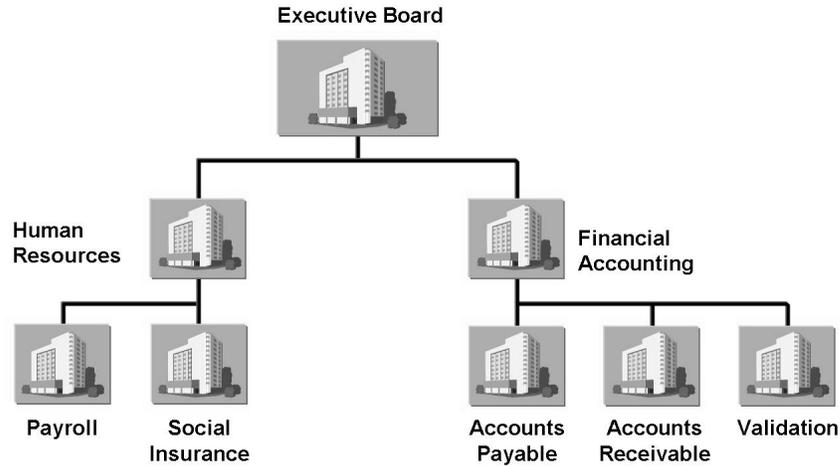


Figure 102: Structural Evaluations

When a structural evaluation is performed, the objects to be evaluated are also listed. Notice that the system interprets these entries differently. The system regards 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

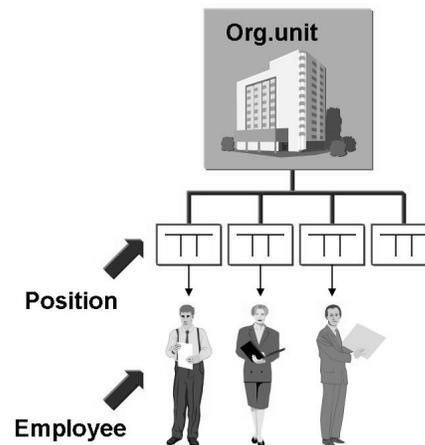
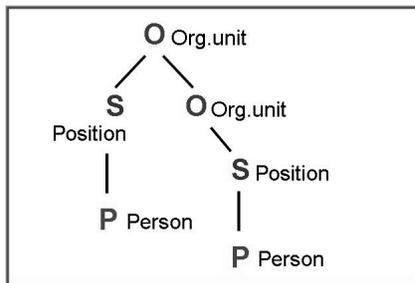


Figure 103: Evaluation Paths

An evaluation path describes a set of relationships between objects in a hierarchical structure. For example, the evaluation path **O-S-P** 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 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.

The report **RHWEGIDO** 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

```

graph TD
    O1[O Org.unit] --> S1[S Position]
    O1 --> O2[O Org.unit]
    S1 --> P1[P Person]
    O2 --> S2[S Position]
    S2 --> P2[P Person]
    
```

Figure 104: Maintaining Evaluation Paths

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.

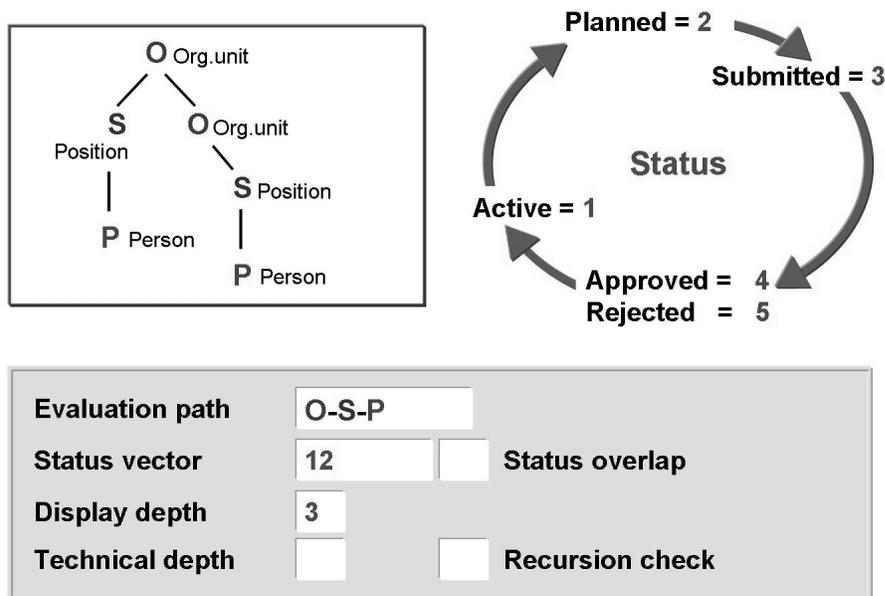


Figure 105: Structure Parameters

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 when all the relationships have been activated internally. During the simulation, all 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. For example, if the value of the field is 3, 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, that is interactive reporting.

Recursion check: If the system detects recursion such as recursive data and 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		
CheckObjTy	Root object	Evaluation path	Status vector
S	C 30015502	A007	

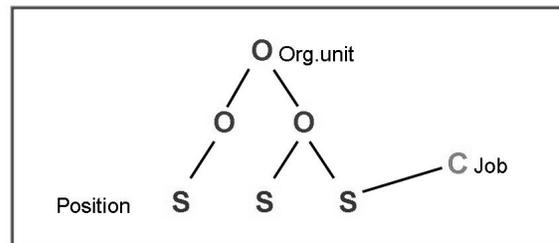


Figure 106: Setting Structure Conditions

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.



Lesson Summary

You should now be able to:

- Identify structure parameters
- Set structure conditions
- Maintain evaluation paths

Lesson: Evaluations-Source Code

Lesson Overview

This lesson helps you create reports for sequential evaluations and structural evaluations. This lesson also describes how to evaluate table infotypes.



Lesson Objectives

After completing this lesson, you will be able to:

- Create reports for sequential evaluations
- Create reports for structural evaluations
- Evaluate table infotypes

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The executive board requires an overview of the staff positions with the names of the employees holding these positions, based on the organizational structure.

Sequential Evaluation



```

TABLES:      OBJEC.
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.

```

Figure 107: Sequential Evaluations - Source Code

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 evaluation or a structural evaluation is carried out. A sequential evaluation is carried out in the example outlined above.

For each selected object, the internal table Pnnnn is filled for each infotype with all the infotype records available. 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-ENDDA contain the data selection period.

When the GET event occurs, all the information from the start of the object period for the infotype 1000 is in the **OBJEC** structure.

Structural Evaluation



```

TABLES: OBJEC, GDSTR.
INFOTYPES: 0002,                "Personal Data
            1003,                "Addresses
            ....

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

```

Figure 108: Structural Evaluations-Source Code

The only difference between a structural evaluation and a 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 indicate an evaluation path before you start 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.

```

Figure 109: Filling Field in the Selection Screen



```

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.

```

Figure 110: Structure Evaluations Without Structure Parameters

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.

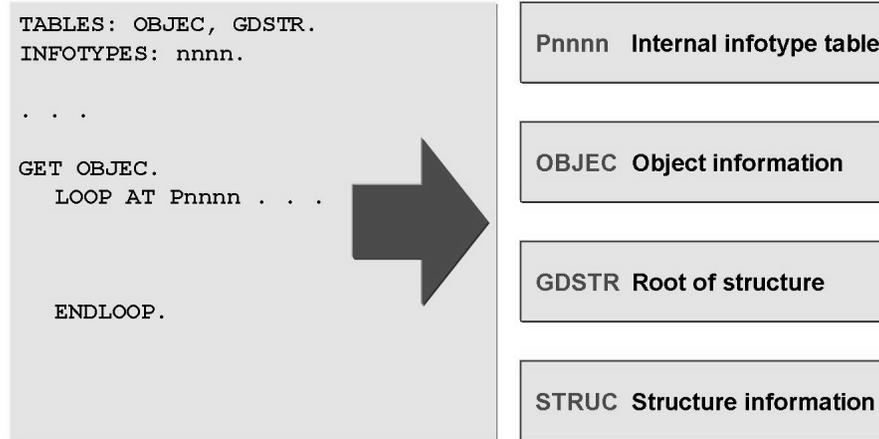


Figure 111: Available Internal Information

For each selected object, the internal table **Pnnnn** is filled for each infotype with all the available infotype records.

For each selected object, the **OBJEC** structure contains the most important information from the infotype 1000 which consists of the plan version, object type, object id, short text, and long text.

The **GDSTR** structure contains the information about 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 the information about the relationship used by the system to select the current object along the evaluation path.

Selection Screen and Additional Data on Relationships



- **Sequential evaluations**

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

- **Structural evaluations**

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

Figure 112: Evaluating Additional Data on Relationships

You must ensure that the relationship type is queried in accordance with the table T77AR before an assignment is effected 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.

Evaluating Infotypes



```

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.

```

Figure 113: Evaluating Table Infotypes

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.

```

Figure 114: Evaluating the Infotype Index

The call sequence of PCH macros facilitates fast object selection using value conditions of infotype fields, such as 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.

```

Figure 115: Reading Infotypes With Function Modules

The RH_READ_INFITY function module can be used to read the Personnel Planning infotype records, such as function group RHDB, in programs without the PCH logical database, such as module pools and 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.

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Exercise 12: Creating a List of Staff Positions

Exercise Objectives

After completing 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

Business Example

The board of directors of your company requires an overview of the staff positions with the names of the employees holding the positions and their telephone numbers.

Task:

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. Take the personal data, such as the last name and first name, from the infotype 0002, and take the telephone numbers from the infotype 0006 subtype 1.

The information pertaining to the staff positions is contained in the position in the infotype 1003. If a position is marked as a staff position, the value “X” is selected for the **P1003-STABS** field.

Process the 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.

In one row, enter only 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.

Continued on next page

The plan version **01**, object type **O**, object id, **30014999**, and evaluation path **O-S-P** should appear as default values.



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

Solution 12: Creating a List of Staff Positions

Task:

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. Take the personal data, such as the last name and first name, from the infotype 0002, and take the telephone numbers from the infotype 0006 subtype 1.

The information pertaining to the staff positions is contained in the position in the infotype 1003. If a position is marked as a staff position, the value "X" is selected for the **P1003-STABS** field.

Process the 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.

In one row, enter only 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.

The plan version **01**, object type **O**, object id, **30014999**, and evaluation path **O-S-P** should appear as default values.



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

```
a) 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'.
```

Continued on next page

```
    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.
```



Lesson Summary

You should now be able to:

- Create reports for sequential evaluations
- Create reports for structural evaluations
- Evaluate table infotypes



Unit Summary

You should now be able to:

- Identify structure parameters
- Set structure conditions
- Maintain evaluation paths
- Create reports for sequential evaluations
- Create reports for structural evaluations
- Evaluate table infotypes



Test Your Knowledge

1. In the _____ field, enter the status values required by the relationship infotype 1001 along the evaluation path so that the appropriate target objects are selected.
Fill in the blanks to complete the sentence.
2. Which of the following ensures that the irrelevant objects, which are the objects that do not meet the structure conditions, are hidden?
Choose the correct answer(s).
 - A AND relationship
 - B OR relationship
 - C Object filter
 - D Branch filter
3. It is an obligatory rule that the convention A = bottom up and B = top down can be taken into account when a relationship is first defined.
Determine whether this statement is true or false.
 - True
 - False
4. Use the _____ statement to determine whether a sequential evaluation or a structural evaluation is carried out.
Fill in the blanks to complete the sentence.
5. Depending on the _____ and _____ parameters, the function module performs an authorization check.
Fill in the blanks to complete the sentence.
6. When structural evaluations take place using the Sequential Selection screen, the evaluation path must be defined in the report.
Determine whether this statement is true or false.
 - True
 - False



Answers

1. In the Status vector field, enter the status values required by the relationship infotype 1001 along the evaluation path so that the appropriate target objects are selected.

Answer: Status vector

2. Which of the following ensures that the irrelevant objects, which are the objects that do not meet the structure conditions, are hidden?

Answer: C

Object filter ensures that the irrelevant objects conditions are hidden.

3. It is an obligatory rule that the convention A = bottom up and B = top down can be taken into account when a relationship is first defined.

Answer: False

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.

4. Use the TABLES statement to determine whether a sequential evaluation or a structural evaluation is carried out.

Answer: TABLES

5. Depending on the AUTHORITY and WITH_STRU_AUTH parameters, the function module performs an authorization check.

Answer: AUTHORITY, WITH_STRU_AUTH

6. When structural evaluations take place using the Sequential Selection screen, the evaluation path must be defined in the report.

Answer: True

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.

Unit 8

Customer-Specific Infotypes

Unit Overview

This unit discusses the infotypes in Personnel Administration and Personnel Planning. It explains how to enhance and create Personnel Administration and Personnel Planning infotypes.



Unit Objectives

After completing this unit, you will be able to:

- Identify the components of an infotype
- Explain standard and customer-specific infotypes
- Enhance Personnel Administration infotypes
- Create Personnel Administration infotypes
- Enhance Personnel Planning infotypes
- Create Personnel Planning infotypes

Unit Contents

Lesson: Overview of Infotypes	224
Lesson: Personnel Administration Infotypes	230
Exercise 13: Creating an Administration Infotype	243
Lesson: Personnel Planning Infotypes	250
Exercise 14: Creating a Table Infotype	257

Lesson: Overview of Infotypes

Lesson Overview

This lesson helps you identify the components of an infotype. In addition, it describes customer-specific infotypes.



Lesson Objectives

After completing this lesson, you will be able to:

- Identify the components of an infotype
- Explain standard and customer-specific infotypes

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The company has data that cannot be stored in standard infotypes. You want to save this data in customer-specific infotypes.

Overview of 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:

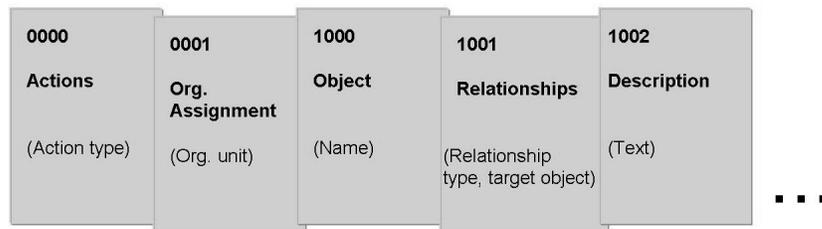


Figure 116: Components of an Infotype

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, each infotype has a module pool that contains an infotype-specific entry, a list screen with validations, and a dialog module.

When you create the individual components, you must note the following factors:

- Specific infotype-dependent objects of infotypes must be edited using the Data dictionary.
- You should edit certain components of infotypes using only 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 factors, you will avoid inconsistencies in infotype-specific objects and table entries. More information about how to create and enhance infotypes is provided in the remaining sections of this unit.



- Where can company-specific information be stored?

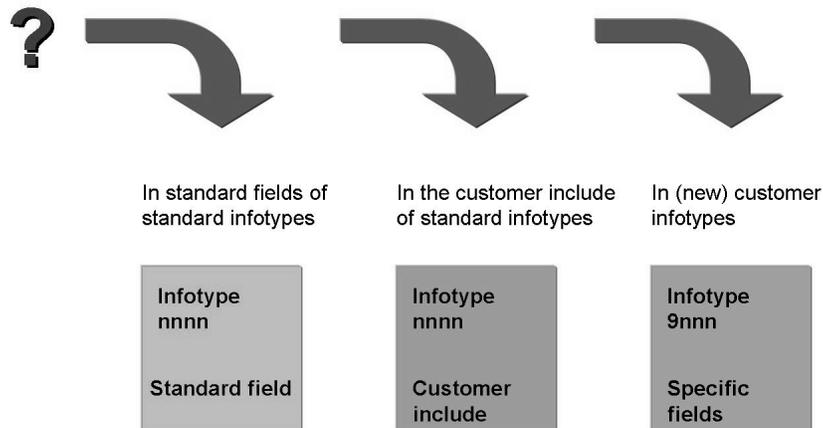


Figure 117: Standard Infotypes/Customer-Specific Infotypes

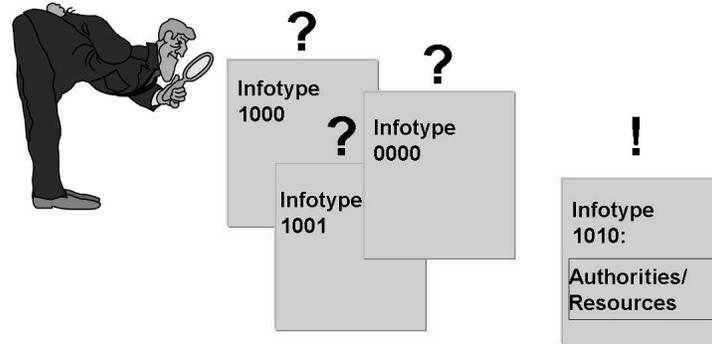
You have three options to create customer-specific information in infotypes:

- You can use a standard infotype with the fields available. For example, you can store the required information in a specific field of a standard infotype.
- You can enhance a standard infotype. This means 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

Figure 118: Using Standard Infotypes

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 want to store the information pertaining to the “Position entitled to the company car”.

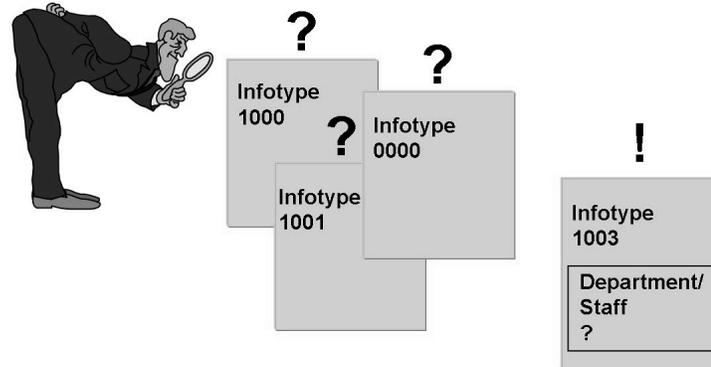
You can use the standard infotype, 1010 (Authorities/Resources), to do this.



Note: You can retrieve information about 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

Figure 119: Enhanced Standard Infotypes

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.

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 want to store the information that this organizational unit is managing a project.

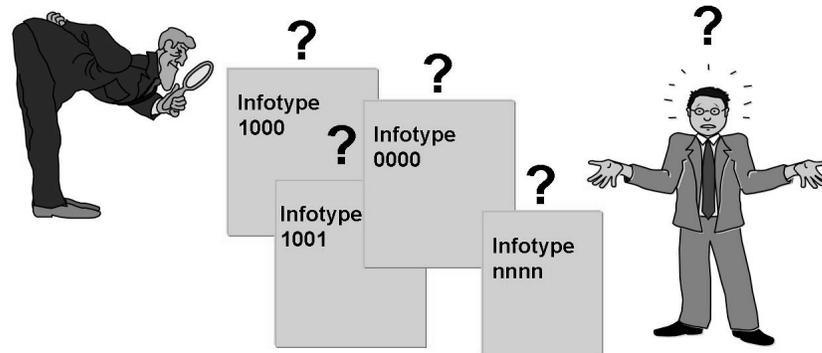
You use the customer include of the standard infotype, 1003 (*Department/Staff*), and add the *Project Management* field.

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 require only minimal change.



- When can you use customer-specific infotypes?



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

Figure 120: Customer-Specific Infotypes

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.



Lesson Summary

You should now be able to:

- Identify the components of an infotype
- Explain standard and customer-specific infotypes

Lesson: Personnel Administration Infotypes

Lesson Overview

This lesson explains how to enhance and create Personnel Administration infotypes.



Lesson Objectives

After completing this lesson, you will be able to:

- Enhance Personnel Administration infotypes
- Create Personnel Administration infotypes

Business Example

All the components of the HR system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The company's personnel department wants a record of the home work centers for individual employees. For this, you need to enhance and create Personnel Administration infotypes.

Enhancing Personnel Administration Infotypes



The diagram illustrates the process of enhancing a Personnel Administration Infotype. It shows two screenshots of the 'Contract Elements (infotype 0016)' form. The left screenshot shows the standard form with fields for 'Contractual regulations', 'Payment period from beginning of illness', and 'Periods'. The right screenshot shows the enhanced form, which includes an 'Additional fields' section with three 'Customer-specific' input fields. A large curved arrow points from the left screenshot to the right one, and a smaller arrow points to the 'Customer-specific 3' field in the right screenshot.

Figure 121: Enhancing Personnel Administration Infotypes - Overview

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, the fields are handled as standard SAP fields in dynamic actions and when you create 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 not covered by the enhancement concept:

- Actions infotype (0000)
- Additional Actions infotype (0302)
- Time Management infotypes (2nnn)
- Object infotype (1000)
- Relationships infotype (1001)

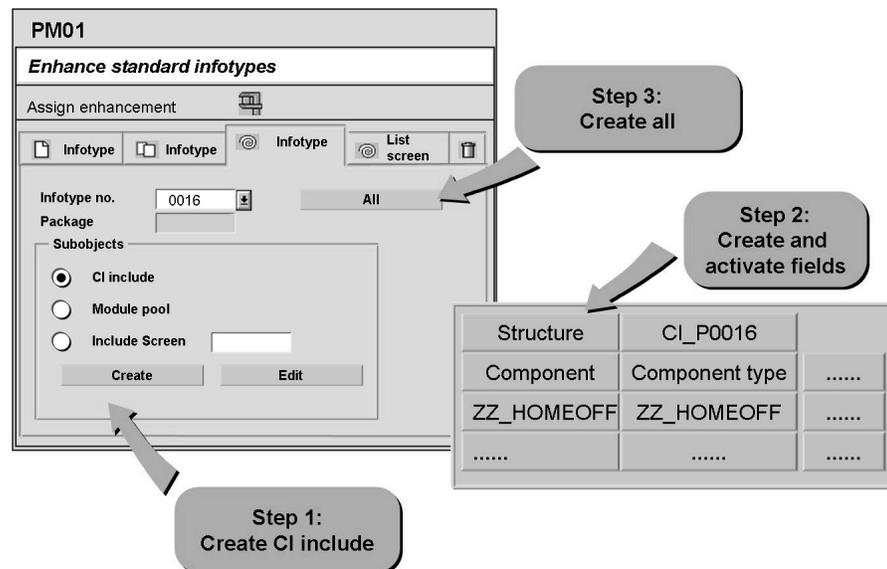


Figure 122: Enhancing Personnel Administration Infotypes - Procedure

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 want to enhance. When you enter the infotype number, remember to enter leading zeros as well.

In the *Subobjects* groupbox, select *CI include*, and click *Create*. The screen *Dictionary: Initial Screen* appears.

Create the CI include and click *Activate*. If you want to add completely new fields, you must 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

Figure 123: Enhancing Infotypes - Created Objects

The include screen is assigned to the standard screen. To do this, click the *Assign enhancement* button. The entries in the table T582C ensure that the screen below the standard fields appears.

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.

Creating Personnel Planning Infotypes



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

Create Edit

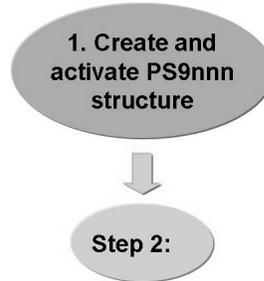


Figure 124: Creating Personnel Administration Infotypes - Step 1

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 All

Package

Employee infotype
 Applicant infotype
 Both

Subobjects

PS structure
 Module pool
 Screen
 User Interface

Create Edit

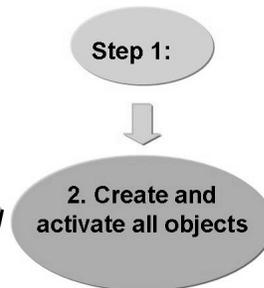


Figure 125: Creating Personnel Administration Infotypes - Step 2

Step 2: After you have created the PS9nnn structure, select *Create All* in the transaction PM01.

You create a new infotype.

- When you do this, the infotype copier automatically creates the following objects:
 - Database table PA9nnn
 - Structure P9nnn
 - Module pool: MP9nnn00
 - Three screens (1000, 2000, and 3000) for the module pool
 - Interface status
 - Dialog module (RP_9nnn)
- When you create new infotypes, the following table entries are automatically created:
 - T777D Infotype check table
 - TDCD Dialog modules



Infotype		9001	Company carpark
General attributes			
Time constraint	2	<input type="checkbox"/> Subtype obligatory	<input type="checkbox"/> Accntng/log.data
Time cnstr. tab.		Subtype table	<input checked="" type="checkbox"/> Text allowed
Maint.af. 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
Display and selection			
Select w/start	3	Create w/o strt	1
Select w/ end	5	Create w/o end	1
Select.w/o date	2	<input type="checkbox"/> List time per.	<input checked="" type="checkbox"/> Choose data
Retroactive accounting trigger			
<input type="checkbox"/> Before ERA date		<input type="checkbox"/> Entry of RA limit time	Past entry all. <input checked="" type="checkbox"/> x
Retr.acct.payr.	<input type="checkbox"/>	Retr.acct.PDC	<input type="checkbox"/>
Technical data			
Single screen	2000	Dialog module	RP_9001
List screen	3000	Structure	P9001
<input type="checkbox"/> List entry		Database table	PA9001
		Applicant infotypes	
		Applicant DBTab	

Figure 126: Creating Personnel Administration Infotypes - Step 3

When you create an infotype, the system does not automatically create the table entries that describe the characteristics of the infotype, tables T582A and T582S. For this reason, you must create the corresponding entries manually in the tables. You use the view V_T582A to maintain the tables. This view appears after you carry out the *Create All* step.

You can maintain the tables at any time in the transaction PM01 on the *Create infotype* screen. To do this, select *Infotype characteristics*.

To create new entries:

- Select *New Entries*, or
- Copy 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



Select *Create All* in the transaction PM01 to create the objects outlined above.

You often have to modify the *Layout* and *Flow logic* components on the screens.

Normally, you 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.

```

Figure 127: Include MPnnnn10

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.

```

Figure 128: Include MPnnnn20

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, you can 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 one time 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, such as 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 the 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. Notice that 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.
  
```

Figure 129: Flow Logic of the Single Screen - PBO

You can carry out infotype-specific initializations in the **P9nnn** PBO module. For example, you can 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.

```

Figure 130: Flow Logic of the Single Screen - PAI (1)

The **INPUT_STATUS** PAI module must be executed if the user makes an entry in a screen field. For this reason, 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.

```

Figure 131: Flow Logic of the Single Screen - PAI (2)

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, you select *Exit*.

After 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. You cannot 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 whether the start date of the infotype record is before the record's end date.

All the fields that appear on the single screen must be listed in the chain. The fields that are only displayed, such as 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>	001
<i>Change</i>	002
<i>Add and copy</i>	004
<i>Delete</i>	008
<i>Lock/Unlock</i>	010

Figure 132: Function-Dependent Screen Control (1)

You determine the attributes of the individual screen fields when you create single screens and list the screens with the ABAP Screen Painter. Notice that the same screen is always used for different functions such as the tasks to create, display, maintain, and delete infotype records. As a result, when you maintain the screen, you cannot indicate whether or not a screen field is to be ready for input. Depending on the organizational data of the employee, certain screen fields may need to be hidden.

Some attributes can only be specified at runtime, and cannot be specified in general. The layout of the screen will change, depending on the function a user has chosen or 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>	200
<i>Display in list screen and change in list screen</i>	400
<i>Add and copy</i>	800

Figure 133: Function-Dependent Screen Control (2)

The value of the *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.

Example:

You want a screen field to be ready for input for the Add and Change functions. For this reason, you maintain the value **006** in the *Modification group 1*. You want one screen field to be ready for input for all functions. In this case, you maintain the value **00F** in the *Modification group 1*.

The **BEGDA** and **ENDDA** fields are normally ready for input for all actions, except for the display of records. For this reason, you assign the value **00E** to the modification Group 1 attribute of these fields.

The *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.

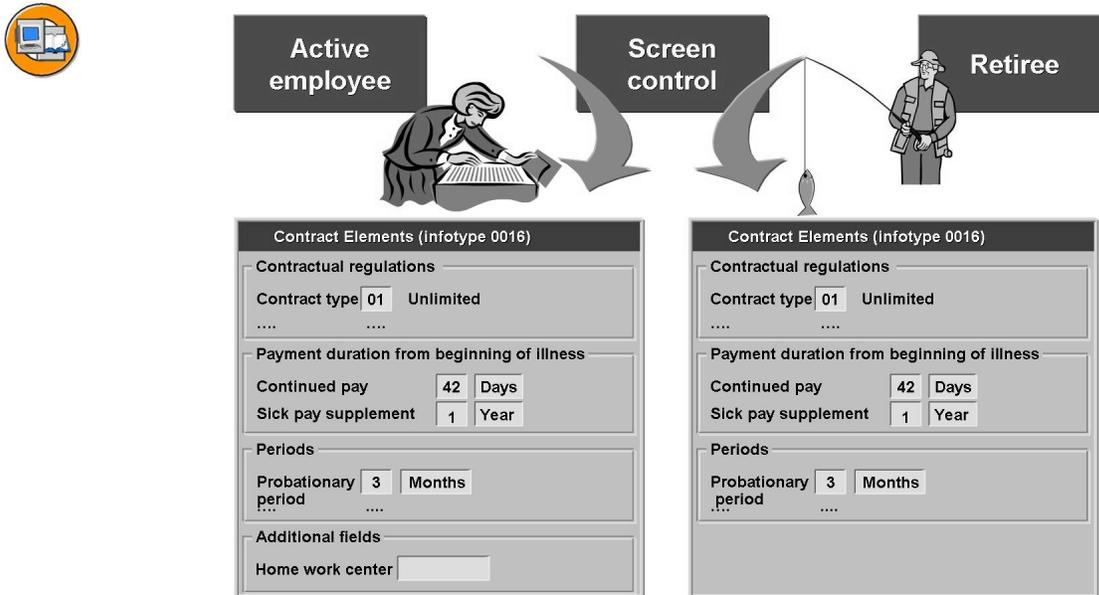


Figure 134: Control-Data-Dependent Screen Control (1)

Along 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

Normally, screen control can be carried out based on the organizational data for the employee or the subtype of the infotype record. To this end, maintain the value of *the 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 the *Modification group 3*. Use the same value for the screen fields that are to be modified in the same way. The same value is used for an I/O field as for the corresponding keyword and any long text that is displayed. Assign the screen fields that are not modified using T588M to the value **SPACE** in the *Modification group 3*.

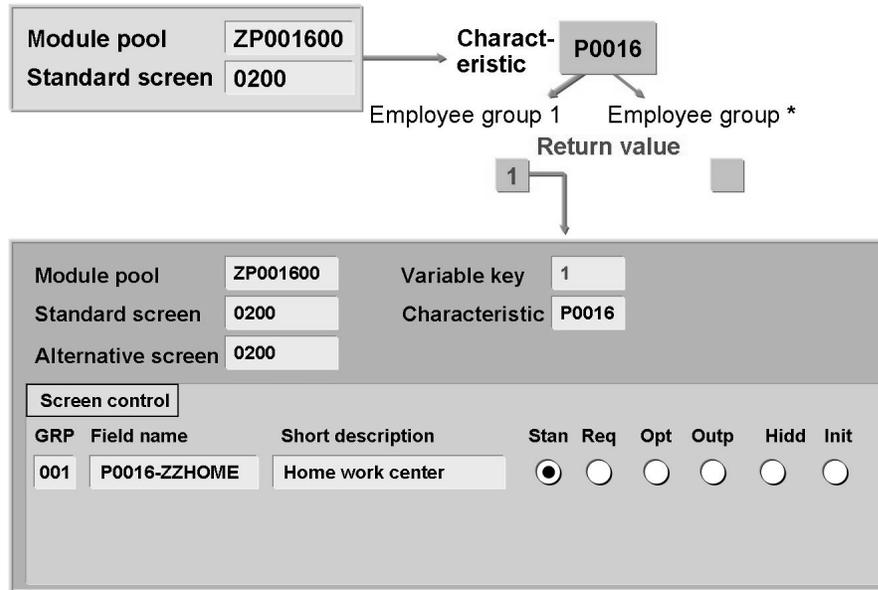


Figure 135: Control-Data-Dependent Screen Control (2)

In the table T588M, you determine which alternative screens you want 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 decision tree of the characteristic, the entry for which the *Variable key* field is empty, which is the basic entry (basic screen), is automatically valid. If no characteristic is available for the corresponding infotype, you must create a characteristic.

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 also valid for the alternative screen.

Exercise 13: Creating an Administration Infotype

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The personnel department wants a record of home work centers for individual employees.

Task 1:

Create the *home work center infotype (##)* infotype for Personnel Administration. Use the infotype number **95##**. (## = group number)

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 the ids for possible equipment for the home work center.
2. Maintain the infotype characteristics (T582A). You can use the characteristics of the infotype 0002 as a template.
3. Convert the *Fulltime* field into a checkbox using the Screen Painter.
4. Test your infotype using the personnel master data maintenance (transaction PA30).

Continued on next page

Task 2:

Enhancement of flow logic of the infotype for the PBO event using a check table for the Equipment field.

1. In your structure **PS95##**, create the foreign key relationship for the **EQUI** field. Enter the **ZZEQUI** table as the check table.

Change the screen for the Detail screen for your new infotype. This enables the corresponding descriptive text to appear next to the Equipment field.

Test your development in HR master data maintenance.

Task 3:

Enhancement of flow logic of the infotype for the PAI event.

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

Solution 13: Creating an Administration Infotype

Task 1:

Create the *home work center infotype* (##) infotype for Personnel Administration. Use the infotype number **95##**. (## = group number)

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 the ids for possible equipment for the home work center.

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

Select *PS Structure* and click *Create*. This automatically takes you to the Data Dictionary structure maintenance screen. In the *Short Description* field, enter the name of your infotype, such as 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 the create mode for the component types. Select *Data element* in the dialog box that appears.

Enter the 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.

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

Continued on next page

2. Maintain the infotype characteristics (T582A). You can use the characteristics of the infotype 0002 as a template.
 - a) Maintain the infotype characteristics (T582A) by copying the characteristics from the 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.
Click F3 to go back to the *Create Infotype* transaction.
3. Convert the *Fulltime* field into a checkbox using the Screen Painter.
 - a) To convert the *Fulltime* field into 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. Choose the *Edit* → *Convert* → *check box* → *left button* . Save and activate the screen.
4. Test your infotype using the personnel master data maintenance (transaction PA30).
 - a) To test the new infotype: Call HR data maintenance, which is the transaction, PA30. Enter an existing personnel number, such as 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*, and *Delete*.

Task 2:

Enhancement of flow logic of the infotype for the PBO event using a check table for the Equipment field.

1. In your structure **PS95##**, create the foreign key relationship for the **EQUI** field. Enter the **ZZEQUI** table as the check table.
Change the screen for the Detail screen for your new infotype. This enables the corresponding descriptive text to appear next to the Equipment field.

Continued on next page

Test your development in HR master data maintenance.

- a) Call the 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 click the *Foreign Keys* icon.

The following foreign key relationship is required:

Check table	Check table field	Foreign key table	Foreign key field
ZZEQUI	MANDT	SYST	MANDT
ZZEQUI	EQUI	PS95##	EQUI

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

```
tables zzequit.
```

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
```

In the layout of the 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*. Next, use the mouse to position the field on the screen.

Continued on next page

Task 3:

Enhancement of flow logic of the infotype for the PAI event.

1. Enhance the flow logic of the Detail screen for your new infotype so that it is possible only to activate the *Fulltime* field if the *Part-time worker* field is not selected in the infotype 0007 *Planned Working Time*. If this field is selected, the error message 855 is to be issued from the RP message class.

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

In the flow logic for the 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.
```

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.
```

2. Test your development in personnel master data maintenance.

- a)



Lesson Summary

You should now be able to:

- Enhance Personnel Administration infotypes
- Create Personnel Administration infotypes

Lesson: Personnel Planning Infotypes

Lesson Overview

This lesson enables you to enhance Personnel Planning infotypes. It also explains how to create Personnel Planning infotypes.



Lesson Objectives

After completing this lesson, you will be able to:

- Enhance Personnel Planning infotypes
- Create Personnel Planning infotypes

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The data on the size and equipment of offices is required for office planning. You need to set up the system so that it is possible to store this data.

Enhancing Personnel Planning Infotypes

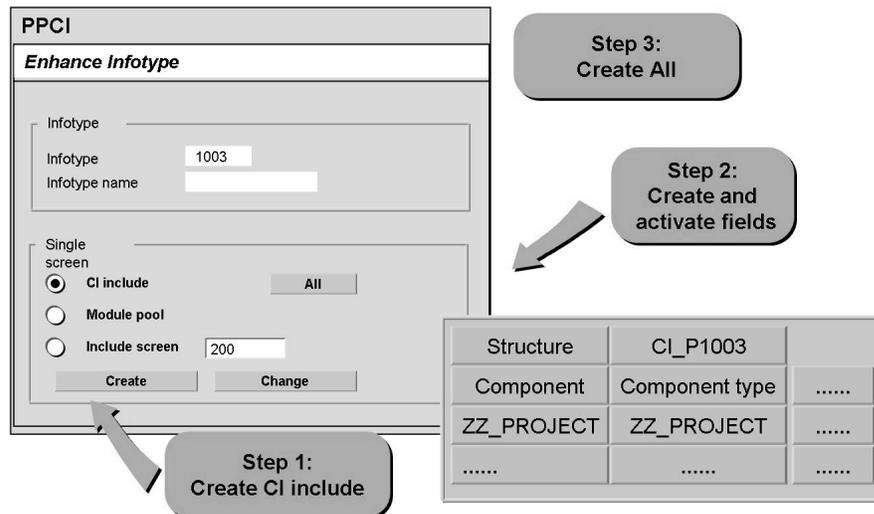


Figure 136: PP Infotype Enhancement

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Start the Personnel Planning infotype copier (transaction PPCI). In the *Infotype* field, enter the four-digit number of the infotype you want to enhance. Select *Extend*.

The *Enhance Infotype* screen appears. Click *Create All*. The *Dictionary: Initial Screen* appears.

Create the include and click *Activate*. If you want 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:

- You cannot enhance the infotypes, 1000 (*Object*) or 1001 (*Relationships*).
- Before you enhance infotypes using SE16, check whether the MAINT field is filled in the table T777I. If this is the case, the infotype cannot be directly maintained and you cannot add an enhancement.

Creating Personnel Planning Infotypes

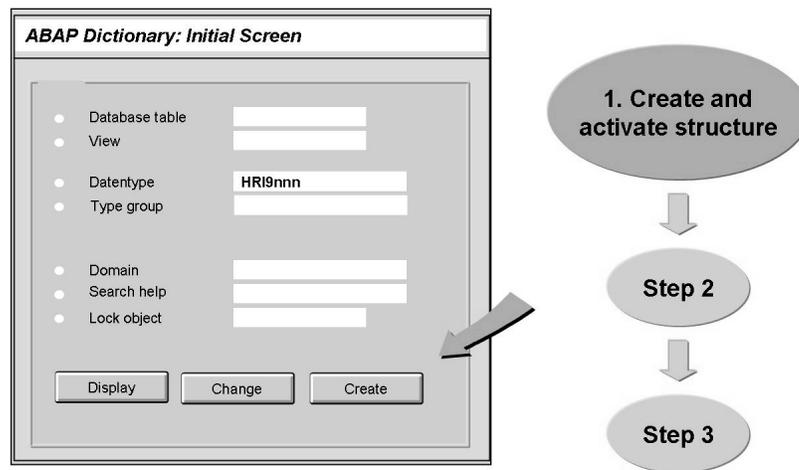


Figure 137: Creating Personnel Planning Infotypes - Step 1

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 the data of your infotype has a repetitive structure, which means 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.



Create Infotype

Infotype

Infotype

Infotype name

Infotype type

- Field infotype
- Table infotype
- Language-dependent infotype
- Country-specific infotype

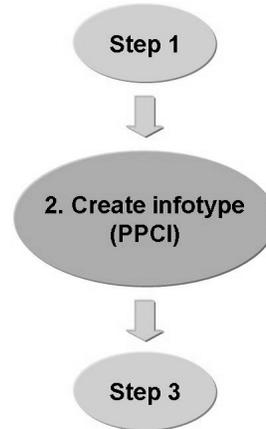


Figure 138: Creating Personnel Planning Infotypes - Step 2

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 click *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
 - Three 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

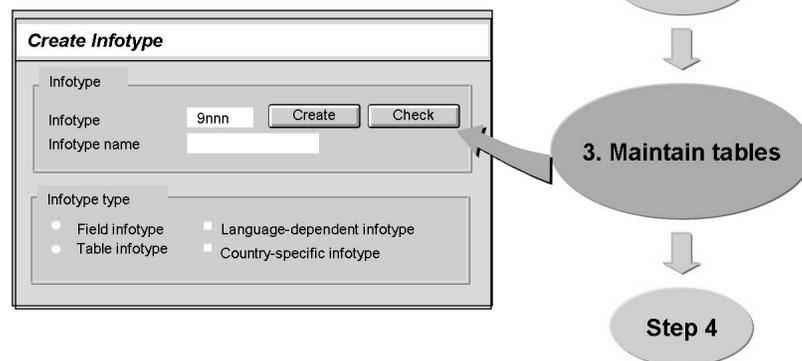


Figure 139: Creating Personnel Planning Infotypes - Step 3

After you create the infotype, entries must be made in the following tables:

- T777I Infotypes for each object type
- T777Z Time constraint of infotypes
- T77CD Customer-specific infotype settings
- T77NI Country-specific infotypes

Step 3: Select *Check*.

Position the cursor on the table you want to maintain and select *Change* on the next screen.

You can also open the tables in Customizing (path: *Personnel Management* → *Organizational Management* → *Basic Settings* → *Data Model Enhancement* → *Infotype Maintenance*).



Note: You must create the create screen 7000 to maintain the new infotype using the transaction PPOME. To do this, select *Infotype* → *Create subscreen*. The entry in the 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.

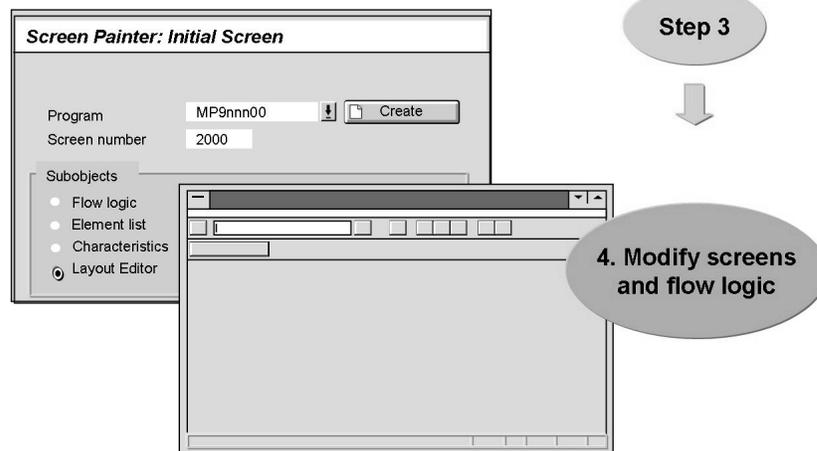


Figure 140: Creating Personnel Planning Infotypes - Step 4

Step 4:

- Modify the new screens to suit your needs.
- To do this, edit the layout, flow logic, and field characteristics (modification groups).
- You can define the infotype-specific check modules in flow logic in the include MP9nnn20.
- The default value 036 is set for the first modification group. When you start the report RHGROUP1 using this value, you establish the functions, such as 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**

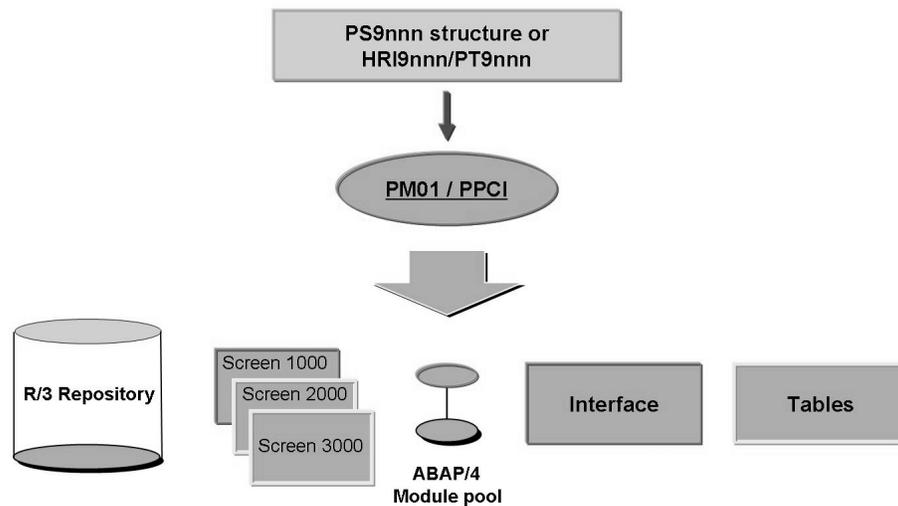


Figure 141: Creating Infotypes - Summary

To sum up: There are three steps to create customer-specific infotypes:

- Define the info-specific fields in the structures, PS9nnn and HRI9nnn, in the dictionary
- Create the central objects of the infotypes using the transactions, PM01 and PPCI (Personnel Administration and Personnel Planning infotype copiers)
- Customer-specific modifications and controls

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Exercise 14: Creating a Table Infotype

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

All the components of the HR system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The data about the size and equipment of offices is required for office planning. Set up the system so that it is possible to store this data.

Task:

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

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, two 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. 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.

Continued on next page

3. Extend the data model of Organizational Management so that you can maintain your new infotype for the object type **O** *Organizational Unit*.
4. Test your new infotype in the expert mode for the organizational unit ## **Controlling**.

Solution 14: Creating a Table Infotype

Task:

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

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, two 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.

- a) 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, two 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 the create mode for component types. Select *Data element* in the dialog box that appears.

Enter the relevant 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. 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.

Continued on next page

Field 2: **EQUI** Equipment, data element: ZEQUI## from the previous exercise.

This field contains the identifier for the equipment for a work center.

a) 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

EQU: You created the data element ZEQUI##

in the exercise on creating an administration infotype.

Create the data element ZSNWRK as outlined in solution 13.

Save and activate the new data element.

Save and activate the structure.

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

Continued on next page

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

- a) 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 the 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 the 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##

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

- a) Test the new infotype in the expert mode in Organizational Management.

Call the 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 click 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 the other functions of this transaction.



Lesson Summary

You should now be able to:

- Enhance Personnel Planning infotypes
- Create Personnel Planning infotypes



Unit Summary

You should now be able to:

- Identify the components of an infotype
- Explain standard and customer-specific infotypes
- Enhance Personnel Administration infotypes
- Create Personnel Administration infotypes
- Enhance Personnel Planning infotypes
- Create Personnel Planning infotypes

Internal Use SAP Partner Only

Internal Use SAP Partner Only



Test Your Knowledge

1. In addition to the structures and database tables, each infotype has a _____ that contains an infotype-specific entry and a list screen with validations, and a dialog module.

Fill in the blanks to complete the sentence.

2. Before you enhance standard infotypes or create new infotypes, you need not perform any check for the infotypes available in the standard system.

Determine whether this statement is true or false.

- True
 False

3. Enhancements are classed as modifications.

Determine whether this statement is true or false.

- True
 False

4. When you create an infotype, the system does not automatically create the table entries that describe the characteristics of the infotype, such as the tables T582A and T582S.

Determine whether this statement is true or false.

- True
 False

5. You use the _____ screen for PP infotype enhancement.

Fill in the blanks to complete the sentence.

6. Which objects are automatically created by the infotype copier when you create a new infotype?



Answers

1. In addition to the structures and database tables, each infotype has a module pool that contains an infotype-specific entry and a list screen with validations, and a dialog module.

Answer: module pool

2. Before you enhance standard infotypes or create new infotypes, you need not perform any check for the infotypes available in the standard system.

Answer: False

Before you enhance standard infotypes or create new infotypes, always check whether the infotypes available in the standard system meet your requirements.

3. Enhancements are classed as modifications.

Answer: False

Enhancements are not classed as modifications. Adding fields to an infotype in the SAP standard system does not cause any problems during an upgrade.

4. When you create an infotype, the system does not automatically create the table entries that describe the characteristics of the infotype, such as the tables T582A and T582S.

Answer: True

When you create an infotype, the system does not automatically create the table entries that describe the characteristics, such as the tables T582A and T582S. For this reason, you must create the corresponding entries manually in the tables.

5. You use the Enhance Infotype screen for PP infotype enhancement.

Answer: Enhance Infotype

6. Which objects are automatically created by the infotype copier when you create a new infotype?

Answer: When you create a new infotype, the infotype copier automatically creates the following objects:

- Database table HRP9nnnModule pool MP9nnn00Table indexthree screens (1000, 2000, 3000) for the module pool
- The structure P9nnn and the CUA interface- a dialog module RH_INFOTYP_9nnn

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Unit 9

HCM ABAP Features

Unit Overview

This unit provides an overview of subroutines and selection reports. It covers how to call subroutines using dynamic actions and to call subroutines for specific periods. In addition, it provides an overview of how to select reports for fast entry in the system call features from reports.



Unit Objectives

After completing this unit, you will be able to:

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

Unit Contents

Lesson: Subroutine Calls	270
Exercise 15: Creating a Dynamic Action	273
Lesson: Report Selection	278

Lesson: Subroutine Calls

Lesson Overview

This lesson helps you get an overview of subroutines. You also learn to call subroutines using dynamic actions and access subroutines for specific periods.



Lesson Objectives

After completing this lesson, you will be able to:

- Call subroutines using dynamic actions
- Call subroutines for specific periods

Business Example

All the components of the HR system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The personnel department needs to maintain the HR infotype records with specific operations and specific periods.

Subroutine Calls



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.
```

Figure 142: Calling Reports Using Dynamic Actions

The Dynamic Actions table, T588Z, is a control table that triggers the steps taken when an infotype record is maintained. Such steps include the tasks to maintain another infotype record, perform a routine, or send an e-mail message.

OP: Signifies a user operation where 02 stands for Change, 04 for Insert, and 08 for Delete a record. The values can be added, such as 06 means that an action is performed if the current record has been changed or inserted.

A: Represents an action performed by the system, such as **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 routines defined in the module pool and external routines. In an external routine, the name of the program is stated 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 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 the field, RP50D-DATE1. This date is calculated by subtracting 14 days from the start date of the locked record.



Table of HR subroutines

Symb.name	Start date	End date	Module pool	Module	Type
XEDT99	01.01.1991	31.12.9999	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 . . .

```

Figure 143: Calling Subroutines for Specific Periods

The HR Subroutines table, T596F, enables you to access the various subroutines for specific periods to solve a task defined by a symbolic name.

In the standard system, the symbolic names of subroutines start with the appropriate country indicator.

The symbolic name is a freely defined name, such as XEDT99, which is used to find a table entry. If a table entry is customer-specific, the symbolic name must start with a special character, such as &EDT99.

You first read the table entry valid for a particular period. This contains the name of a program in the field, **T596F-PGMNA**, and the name of a subroutine contained in this program in the field, **T596F-MODNA**. You then use a dynamic action to call this subroutine from your program.

Exercise 15: Creating a Dynamic Action

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

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.

Task:

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*.

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.

Solution 15: Creating a Dynamic Action

Task:

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.

Continued on next page

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*.

- a) To create a dynamic action:

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

Make the following entries in the table:

Info-type	Type	Field	FC	No	A	Variable function part
0034	9xx		04	0		*----- Corporate function -----*
0034	9xx		04	1	F	<i>COMPUTE_DATE(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..
```

Continued on next page

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.
 - a) Test the dynamic action:

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

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



Lesson Summary

You should now be able to:

- Call subroutines using dynamic actions
- Call subroutines for specific periods

Lesson: Report Selection

Lesson Overview

This lesson helps you get an overview of selection reports. You also use selection reports for fast entry in the system and the call features from reports.



Lesson Objectives

After completing this lesson, you will be able to:

- Use selection reports for fast entry
- Call features from reports

Business Example

All the components of the HCM system have been implemented at your company, Training International. The company wants to use the special features of report programming within Human Resources. The personnel department needs the selected records of the master data and the time data to be transferred to the fast entry.

Report Selection



```
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'.
```

Figure 144: Selection Reports for Fast Entry

If you use the fast entry function for the master data and the 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 the reports 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.



```

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.

```

Figure 145: Calling Features from Reports

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, the 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.



Lesson Summary

You should now be able to:

- Use selection reports for fast entry
- Call features from reports



Unit Summary

You should now be able to:

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



Test Your Knowledge

1. OP stands for a user operation, where 02 indicates _____, 04 represents Insert, and 08 signifies Delete a record.

Fill in the blanks to complete the sentence.

2. In the standard system, the symbolic names of subroutines start with which indicator?

Choose the correct answer(s).

- A Country
- B City
- C Customer
- D Personnel

3. If the return value for a feature consists of a table, use the function module, HR_FEATURE_BACKVALUE.

Determine whether this statement is true or false.

- True
- False

4. If you use the fast entry function for the master data and the _____ data, you can use reports to effect an initial selection of personnel numbers.

Fill in the blanks to complete the sentence.



Answers

1. OP stands for a user operation, where 02 indicates Change, 04 represents Insert, and 08 signifies Delete a record.

Answer: Change

2. In the standard system, the symbolic names of subroutines start with which indicator?

Answer: A

In the standard system, the symbolic names of subroutines start with the Country indicator.

3. If the return value for a feature consists of a table, use the function module, HR_FEATURE_BACKVALUE.

Answer: False

If the return value for a feature consists of a table, use the function module, HR_FEATURE_BACKTABLE.

4. If you use the fast entry function for the master data and the time data, you can use reports to effect an initial selection of personnel numbers.

Answer: time



Course Summary

You should now be able to:

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

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Appendix 1

Report Categories and Clusters



- 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.

Figure 146: Appendix



Contents:

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

Figure 147: Appendix



```

TABLES: PCLn.                "Import/export table
INCLUDE: RPCnxy0.            "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.

```

Figure 148: Exporting Data

The cluster definition is included using the INCLUDE statement.

The EXPORT command writes one or more data objects with the xy-KEY to the 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.

```

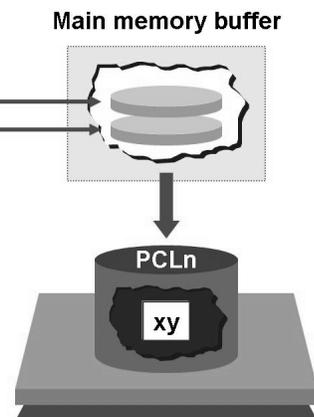


Figure 149: Export Using Buffer

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 the 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.

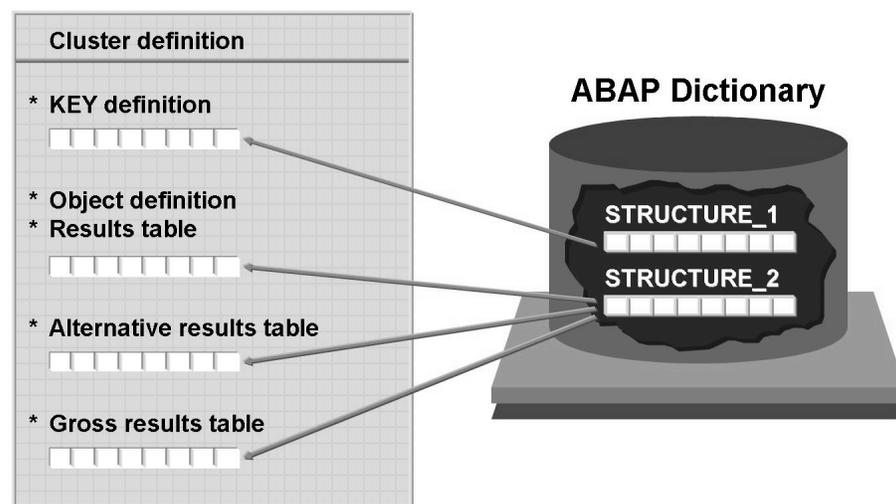


Figure 150: Cluster Definition/Payroll Results (1)

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 HCM country indicator.

The cluster key is contained in the RX-KEY structure.

All the fields in the key and data objects are defined in the ABAP Dictionary structures.

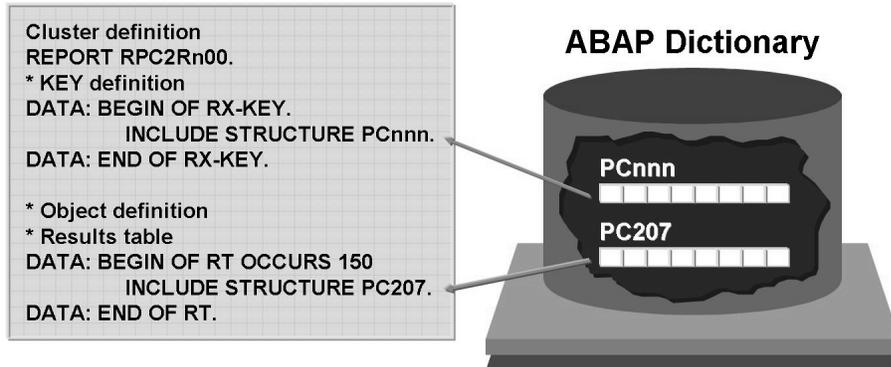


Figure 151: Cluster Definition/Payroll Results (2)

The fields for the key and objects are defined in ABAP Dictionary structures. This means that you can reuse them 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 the 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           = directory
  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.
        
```

Figure 152: Determining the Current Payroll Result

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 indicate the period by entering the payroll period in the selection screen. If you indicate the 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.

Next, you 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.
```

Figure 153: Processing the Cluster Directory

This function module fills the table, EVPDIR (evaluation periods), with the payroll result data from the selected period. The descriptive data from the current result, A, and previous result, P, is written to the table, EVPDIR.

Period parameter: Specifies the period in which the payroll runs for a payroll area. Only one period parameter is assigned to a payroll area. Payroll periods are defined for each period parameter.

In-period: Represents period in which a payroll result is created.

Payroll type: Signifies the indicator for “Type of Payroll Result” according to the 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: Represents the 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

Figure 154: Personnel Planning External Enhancement: Topic Objectives



● **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*)

Figure 155: External Object Types and 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

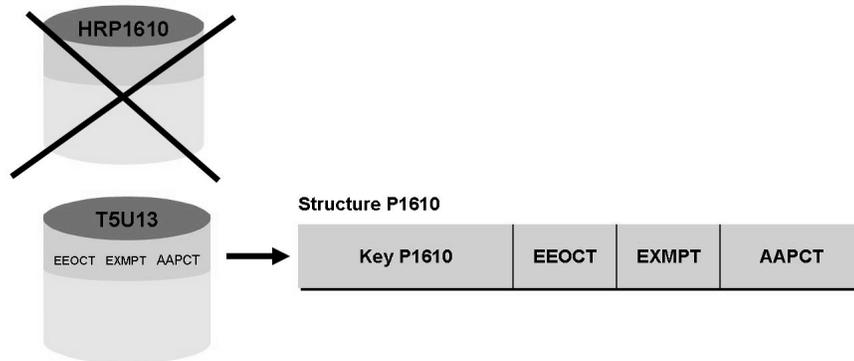


Figure 156: Personnel Planning External Infotypes

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Appendix 2

Introduction to Human Capital Management



Contents:

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

Figure 157: Introduction to Human Capital Management

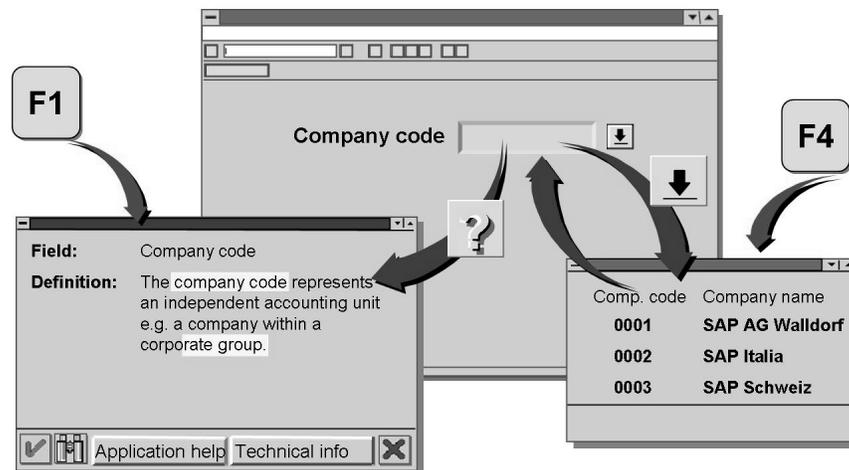


Figure 158: Help on the Screen Fields

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 the glossary information for terms highlighted in the help text.

To do this, point to the term and click the highlighted term.

For further information on possible entries and the field definition, select *Application Help* or *Technical info*.

Access a list of possible entry values or search helps to search for the data you require. To do this, place the cursor on the entry field and press F4 or click 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 click *Choose*, click F2, or point and double-click the left mouse button.

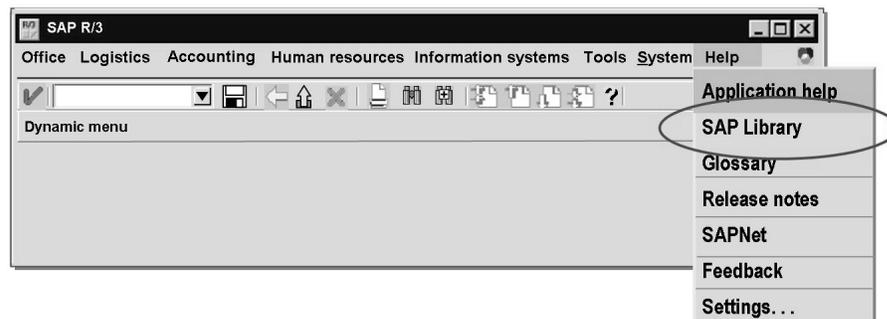


Figure 159: Getting Help with the SAP System

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, such as the tasks to log on to and log off from the SAP system, and work with several sessions.

It describes the common screens.

It describes the 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.

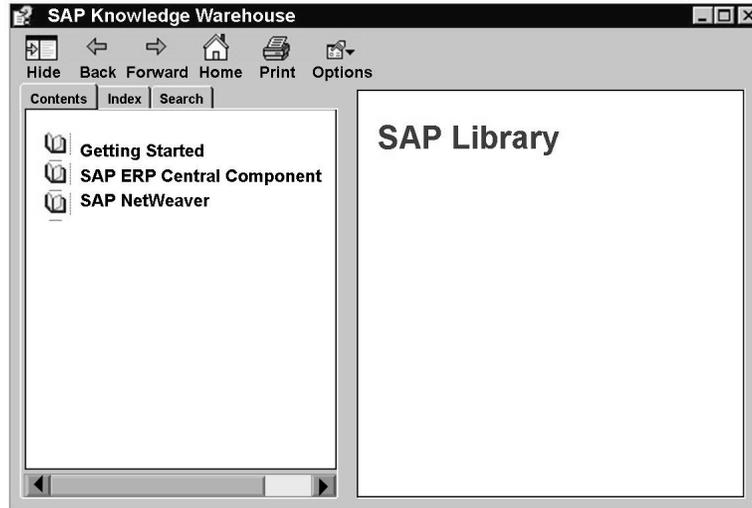


Figure 160: SAP HTML Help Files

Most SAP Help files are HTML files.

The *SAP Library* is an online library of the entire SAP system 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 the technical terms used within the SAP system.

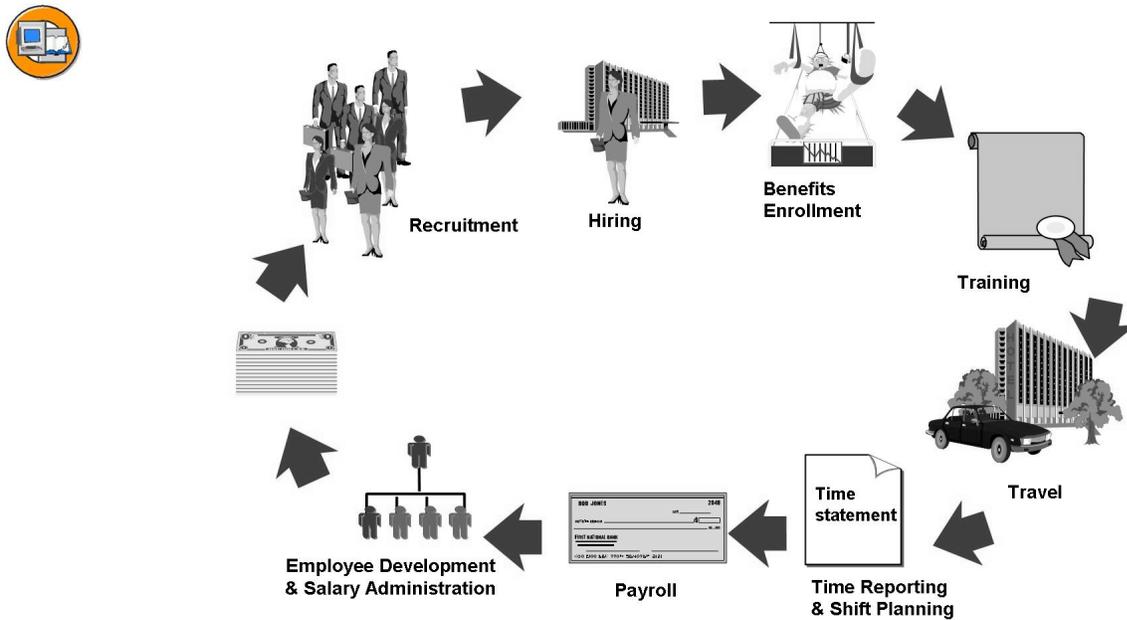


Figure 161: Human Resource Process

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

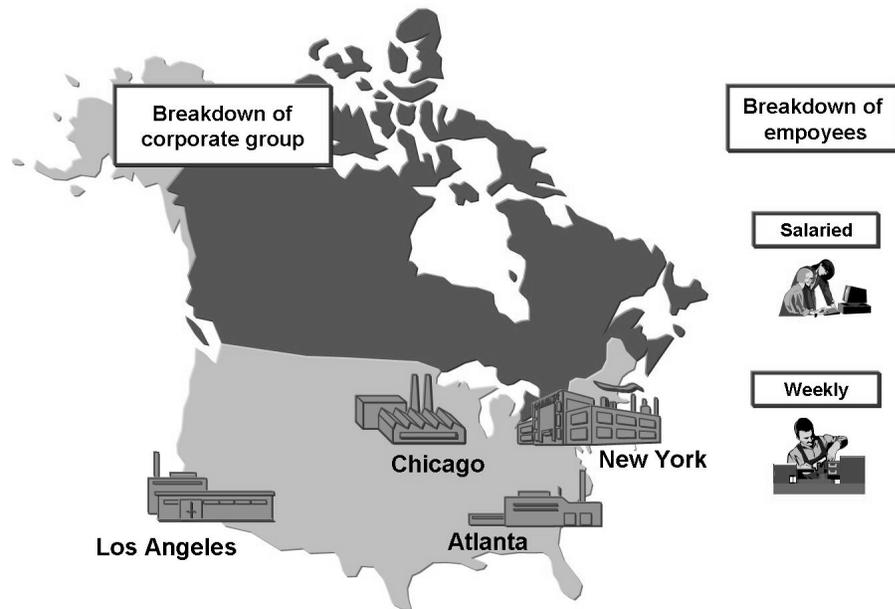


Figure 162: The Structure of the Trainig International Company

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 signifies the division of a corporate group into company code, personnel areas, and personnel subareas.

The first step to record 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, such as active or inactive, and a type, such as salary or hourly.

Employees are also assigned a position within the organization - within employee groups and employee subgroups.

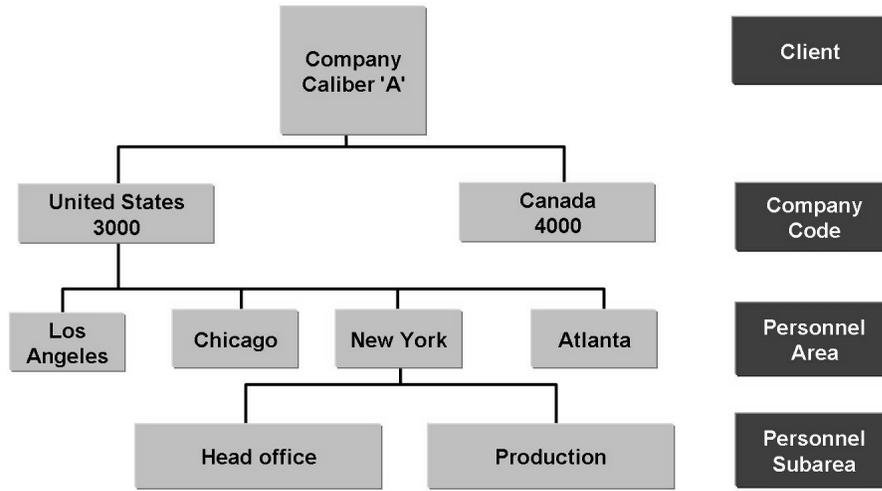


Figure 163: The Enterprise Structure of the Trainig International Company

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 code.

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.

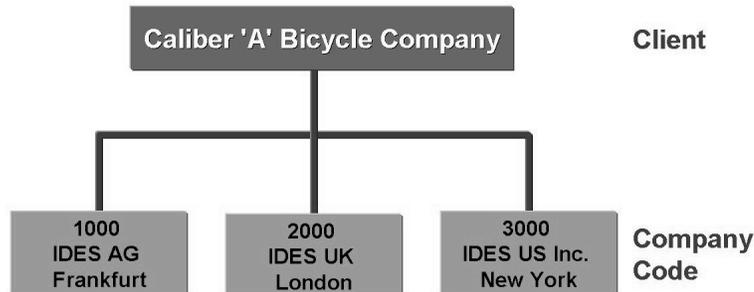


Figure 164: Client and Company Codes

In legal and organizational terms, the client is a self-contained unit within the system. You should take into account the following points before you decide whether or not to set up a client:

You cannot access personnel data in other clients. This protects the data from being accessed by unauthorized users.

You cannot assign access authorization or run evaluations for all clients.

There is no exchange of data among 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, as all the important control information for the Human Resources system is defined at the personnel subarea level.

All of the Trainig Internationa lcompany's United States employees belong to the company code, 3000.



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

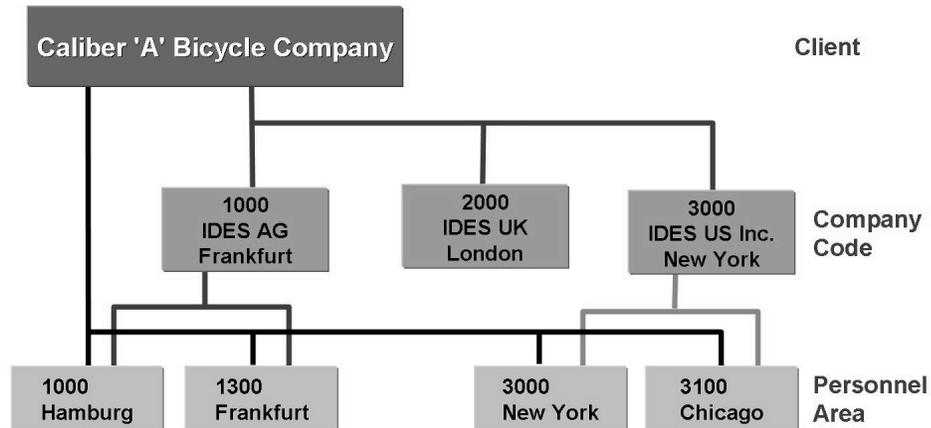


Figure 165: Personnel Area

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 the default values for data entry, such as for the payroll area.
- It is a selection criterion for reporting.
- It constitutes a unit in authorization checks.

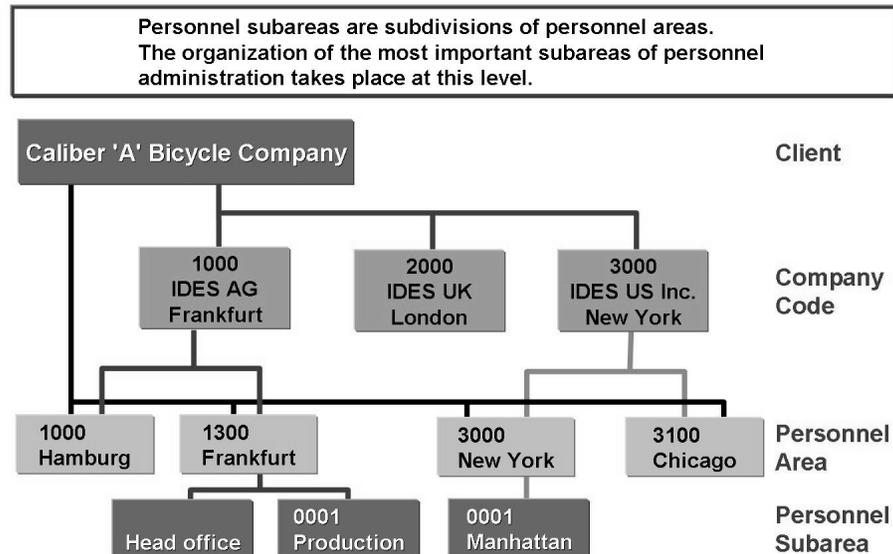


Figure 166: MSI's Personnel Subareas

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. Control features are stored according to the country.

The main organizational functions of the personnel subarea are as follows:

To specify the country grouping. The 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.

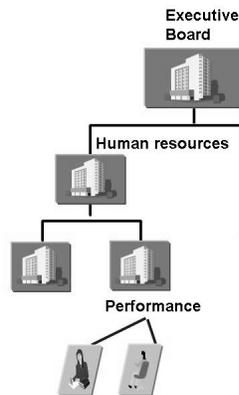
To generate a default pay scale type and area for an employee's basic pay.

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	Administrator
Position	
Percentage	100
Job	50000064
Org.unit	50000827
Org.key	0001

Figure 167: The Organizational Structure of the Trainig International Company

When an employee is hired, the employee is assigned to an organizational unit, a job, and a position in the *Organizational Assignment* infotype, 0001.

The 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.

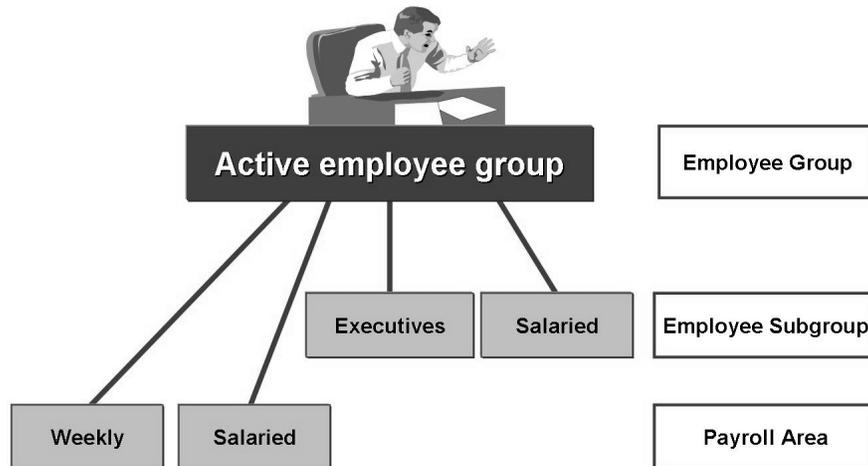


Figure 168: The Administrative Personnel Structure of the Trainig International Company

The administrative personnel structure consists of three elements:

- Employee groups
- Employee subgroups
- Payroll areas

A part of the personnel structure for the Trainig International company is shown above.

The active employees belong to the employee group, Active.

The employee group, Active Employees, is subdivided into different employee subgroups, such as salaried employees, trainees, and monthly-wage earners.

A payroll area groups together employees for whom the payroll is to be run at the same time.

In the new Customer Service Center of the Trainig International 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 semimonthly basis.



Figure 169: The Employee Groups of the Trainig International Company

Employee groups represent the different types of employees in an organization.

The Trainig International company has active employees, retirees, contractors, and external employees.

Employee groups are used:

- To generate default values for the 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.

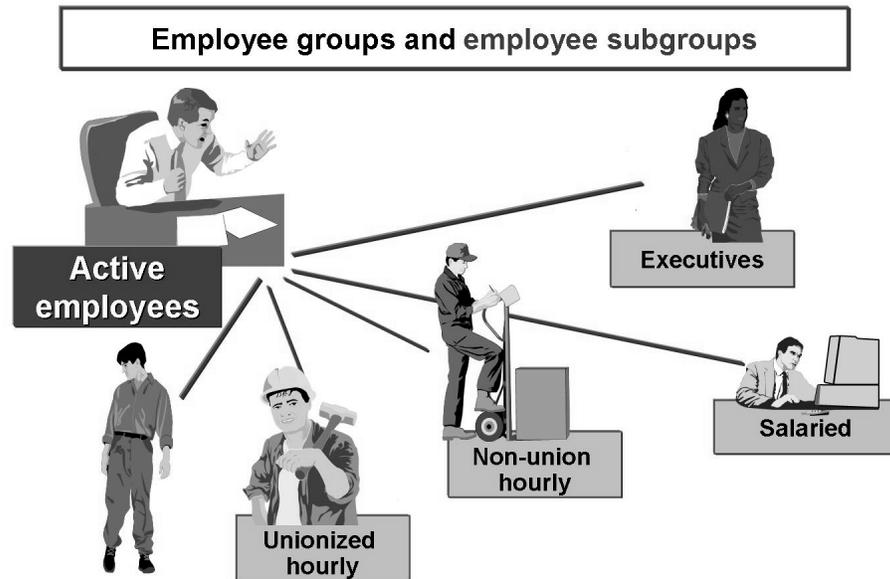


Figure 170: The Employee Subgroups of the Trainig International Company

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)

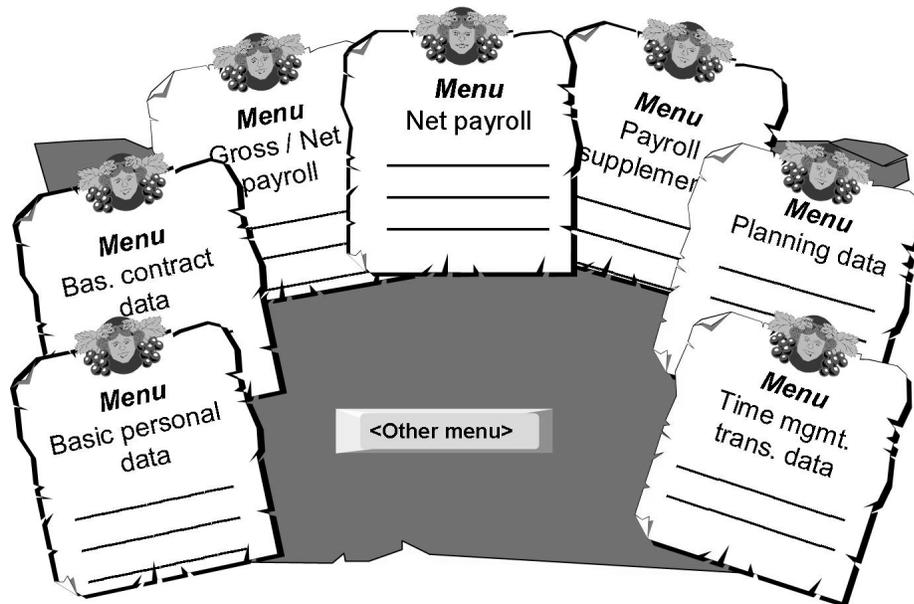


Figure 171: Infotypes

Infotypes are used in HCM 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, but 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. Alternatively, you can use the pull-down menu to find the required menu.

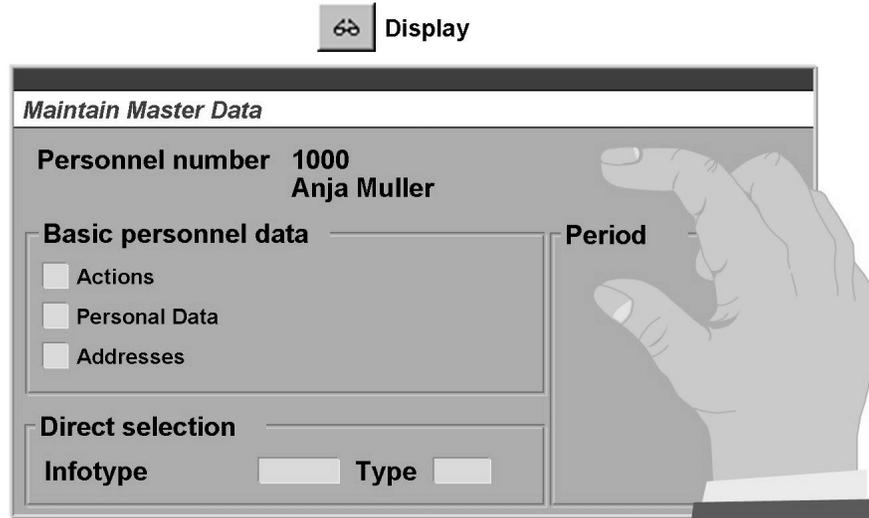


Figure 172: Displaying Employee Information

To access a selection screen with an infotype menu:

1. Select the menu path:
Human resources → Personnel management → Administration → Display master data
2. Enter a personnel number to display the information about the specified employee.

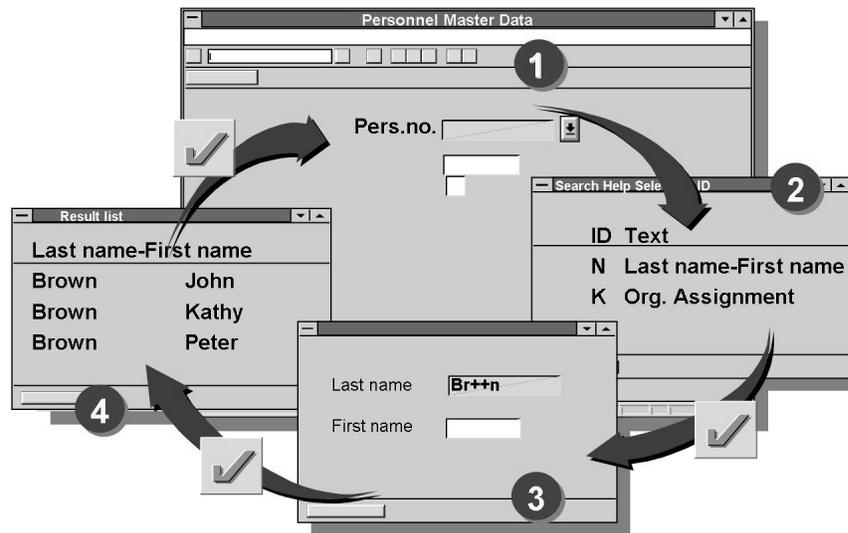


Figure 173: Using Search Helps

There are many search utilities that contain 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 the Possible entries button.

A list that shows all the available search utilities appears. To choose the required matchcode, double-click it.

In the next screen, you enter search terms, such as last and first name.

You will then obtain a list of all the personnel numbers that meet the search criteria. You can choose the personnel number you require from this list.

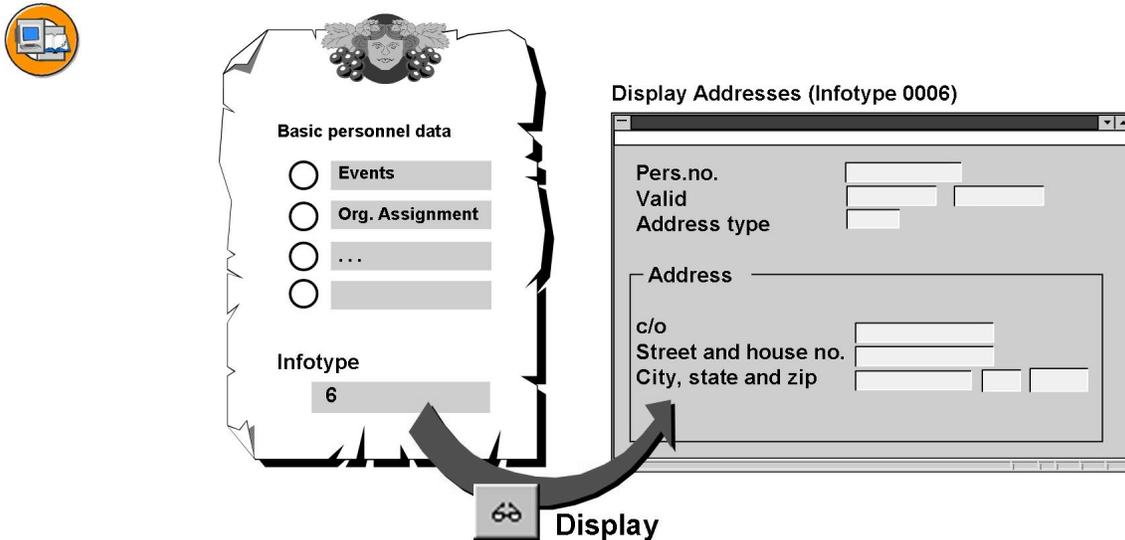


Figure 174: Infotype Selection by Number

If you know the number of the infotype, you can enter it in the Infotype field. Select the requested processing type to make the system 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

Figure 175: Maintaining Infotypes

There are three functions to maintain infotypes: change, create, and copy.

If you use the **Change** function, you can correct a record without the need to create a new one. You choose this function to correct the field contents or add additional data. This function does not create a new record, and for this reason, no history is available.

If the personal data of an employee has changed, select the **Create** function to enter new data. A blank input template appears. 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 shows the current record along with the field contents. You can either copy the field contents that have 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 that relate 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, and 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.

Figure 176: Editing Infotype Records

Use the icons when you edit infotypes.

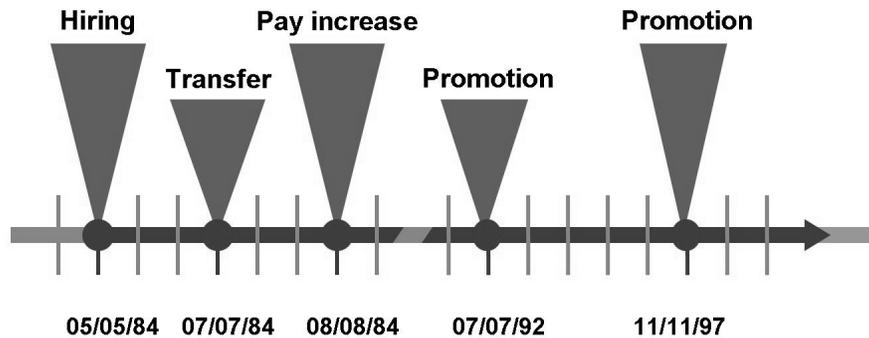


Figure 177: Maintaining Employee Information by Date

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.

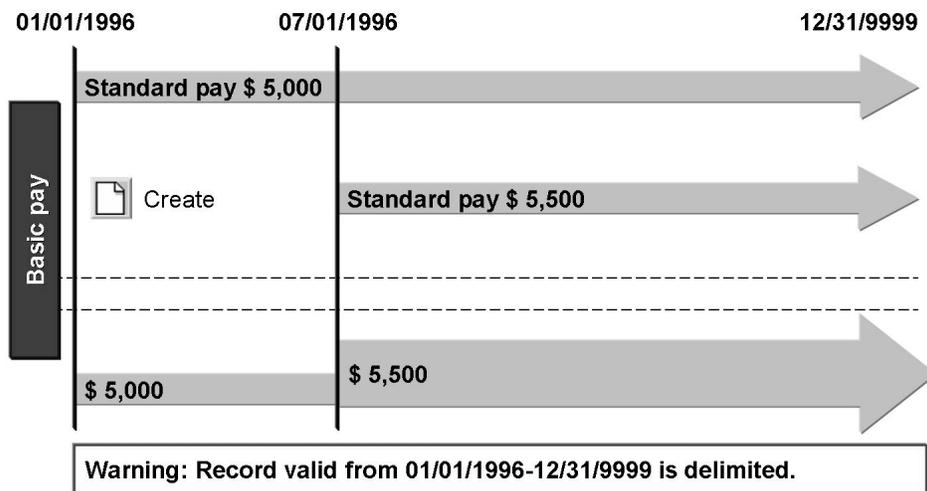


Figure 178: Generating a Data History

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.

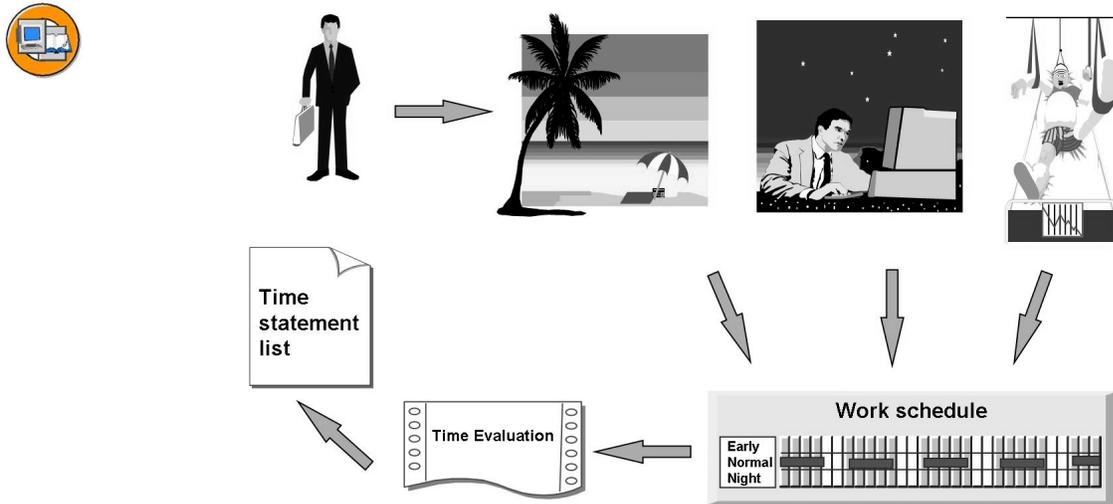


Figure 179: Time Management (1)

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. Exceptions must also be recorded in the system.

The working hours of each employee are evaluated, and the employee is paid accordingly.

A time statement list can be printed for each employee at the end of the month.



Time Management

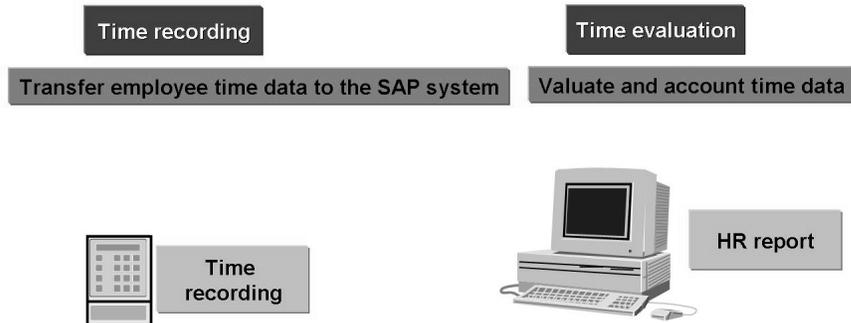


Figure 180: Time Management (2)

Internal Use SAP Partner Only

Internal Use SAP Partner Only

HCM Time Management is concerned 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.

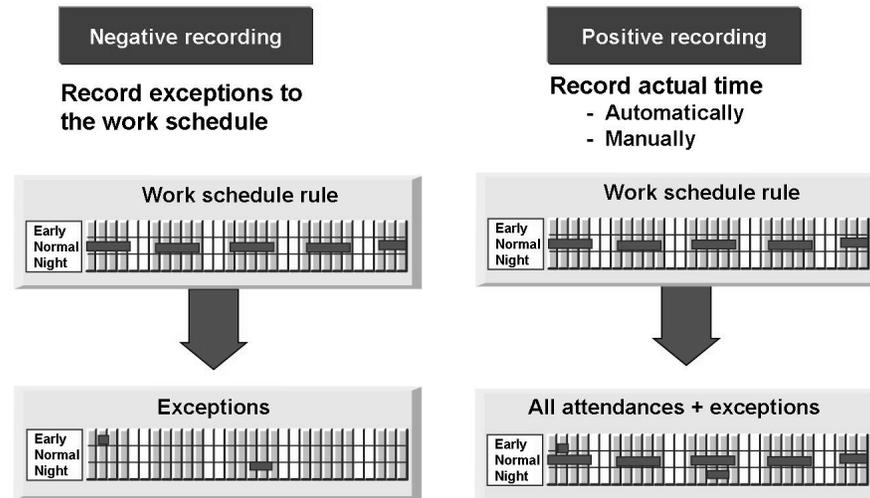


Figure 181: Time Recording

There are two methods to record times:

- Recording exceptions to the work schedule: Only the time data that is different from the specifications in the employee's work schedule is recorded, such as sickness, substitution, and vacation.
- Additional recording of actual times: All the attendance times (actual times) of the employee 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 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.

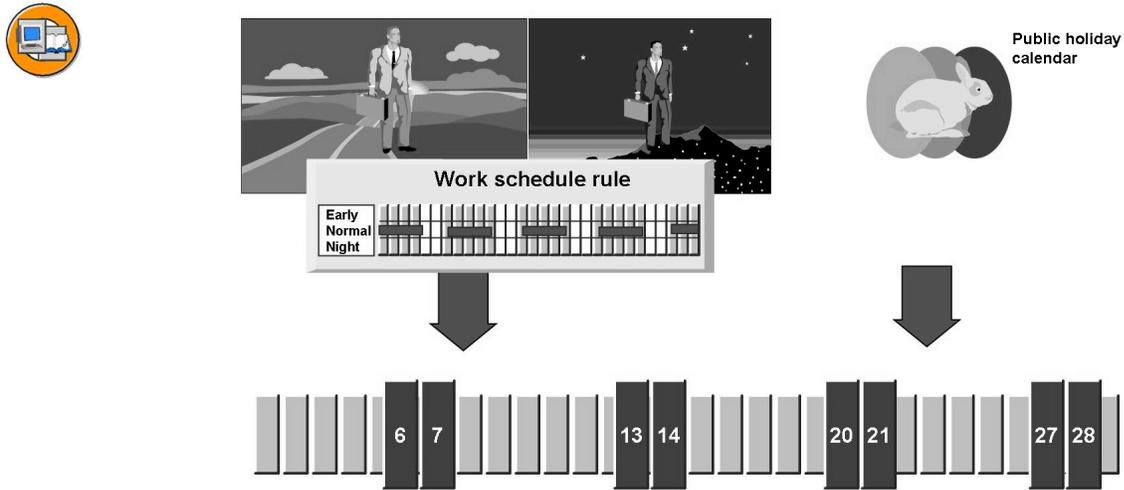


Figure 182: Prerequisites of Time Management

To work with the Time Management system, two items are required:

- A calendar of public holidays
- A work schedule, which specifies the daily and monthly working patterns

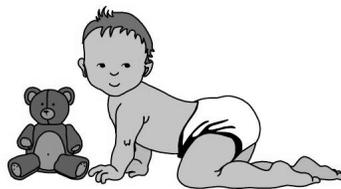


Figure 183: Absences

You can record employee absences in the 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/nonmilitary service

There are various entry screens in the infotype, 2001, depending on whether you are recording a general absence, an absence with quota deduction, or a period of sickness or leave.

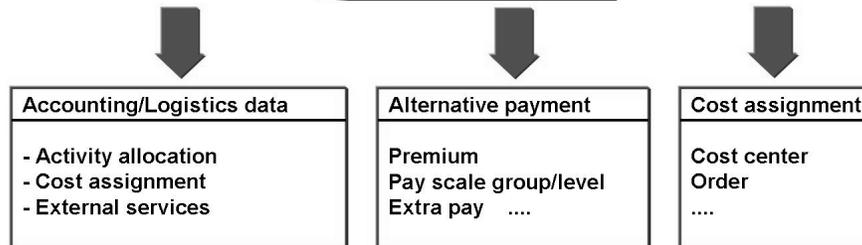


Figure 184: Overtime

Overtime is the time worked in addition to the planned working time defined in the work schedule rule of the employee:

- 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.

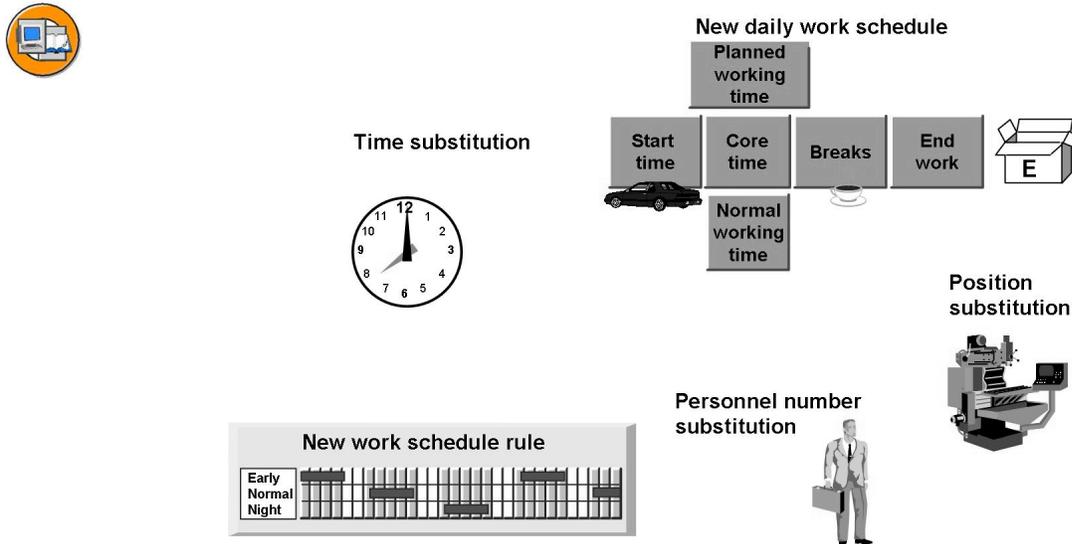


Figure 185: Substitutions

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.

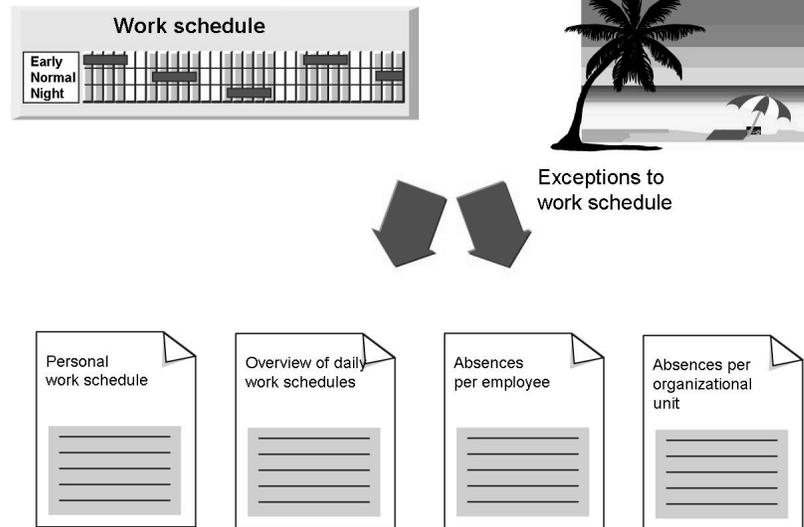


Figure 186: Time Data Reports

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

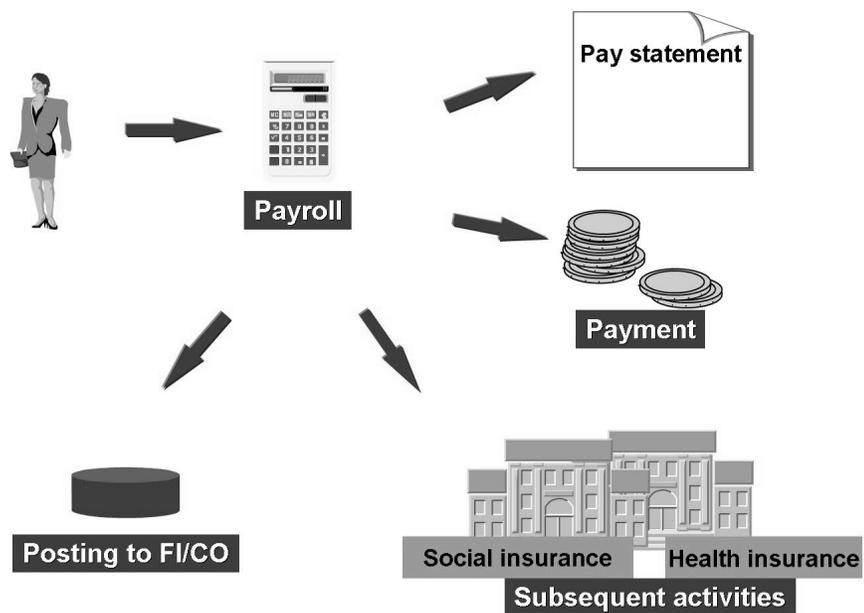


Figure 187: Payroll

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.

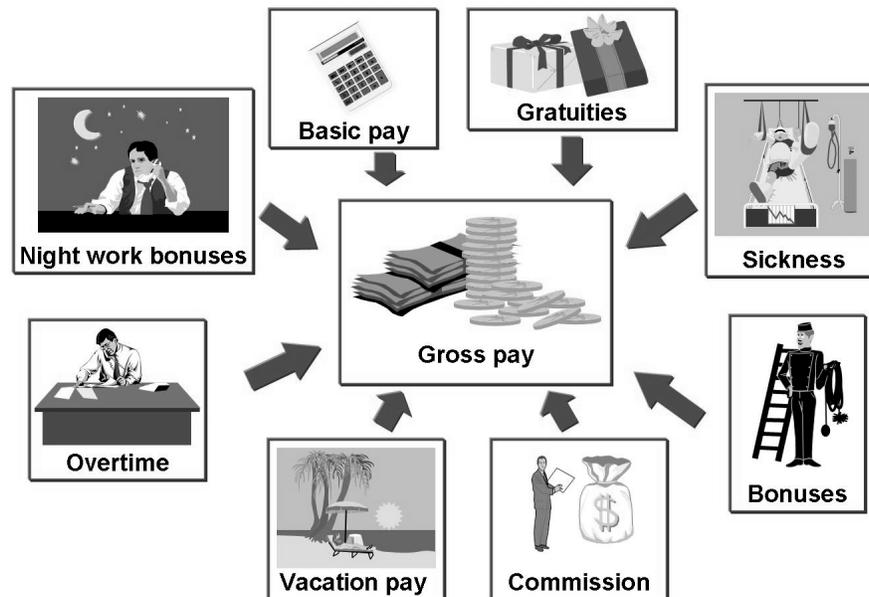


Figure 188: Calculating the Gross Amount

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, and gratuities.

Deductions might include United Way, and a medical or bond deduction. A wage type is a four-digit code that represents an earning or a deduction. Wage types can be entered online or generated during the payroll run using rule tables.

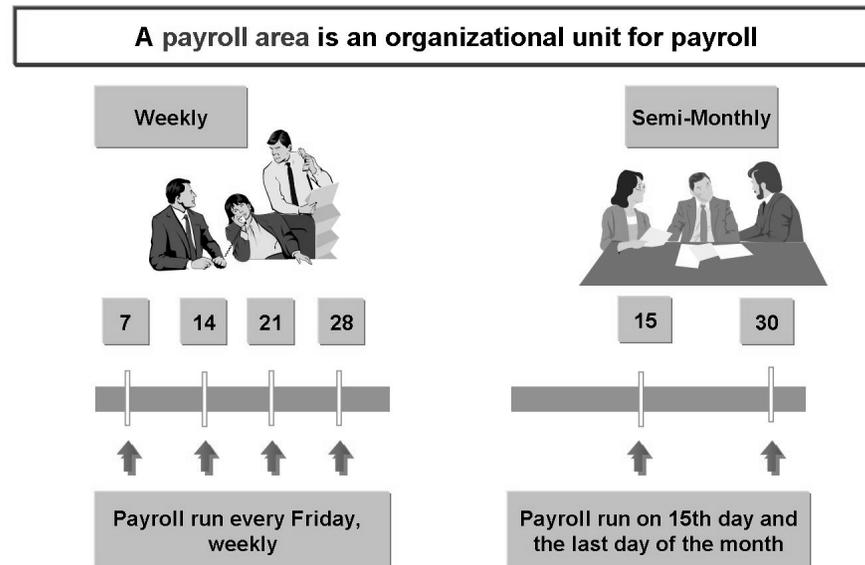


Figure 189: Payroll Areas

The payroll program is normally 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 to run 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.

The 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.

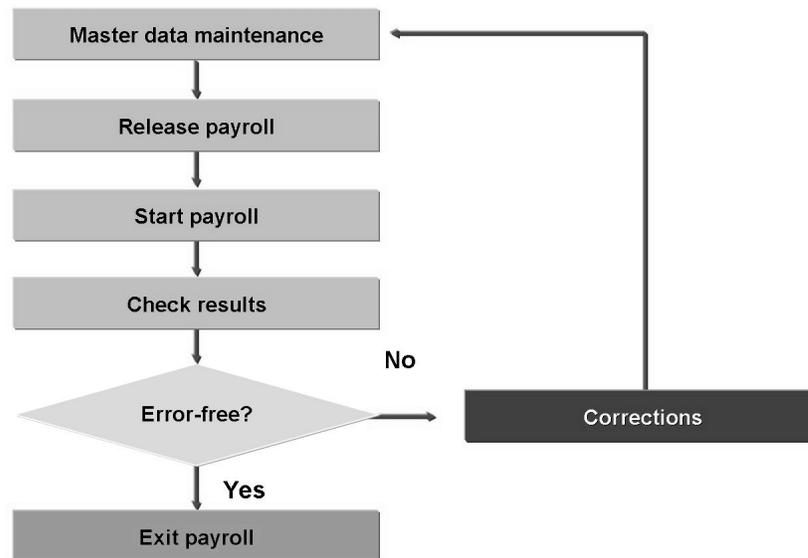


Figure 190: Payroll Control Record

The payroll control record keeps track of each payroll run to ensure that each payroll 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 the master data or the 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, you cannot make any changes that affect the payroll, past or present. You cannot 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 corrections.

The correction status releases the employee's data. You can then enter the changes for the payroll period and 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 the personnel numbers in the selected payroll area.

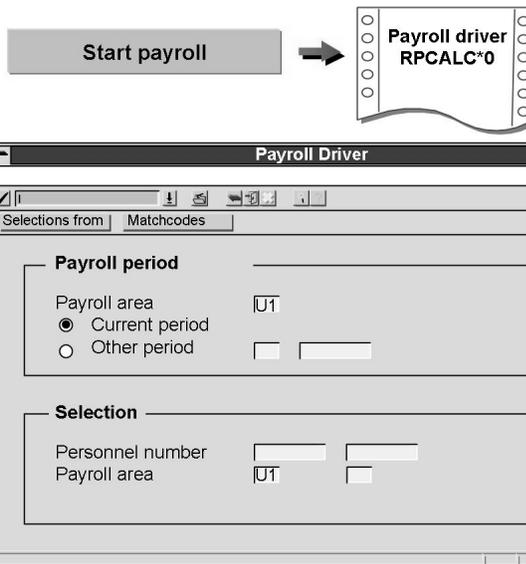


Figure 191: Payroll Program: Selection Screen

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.

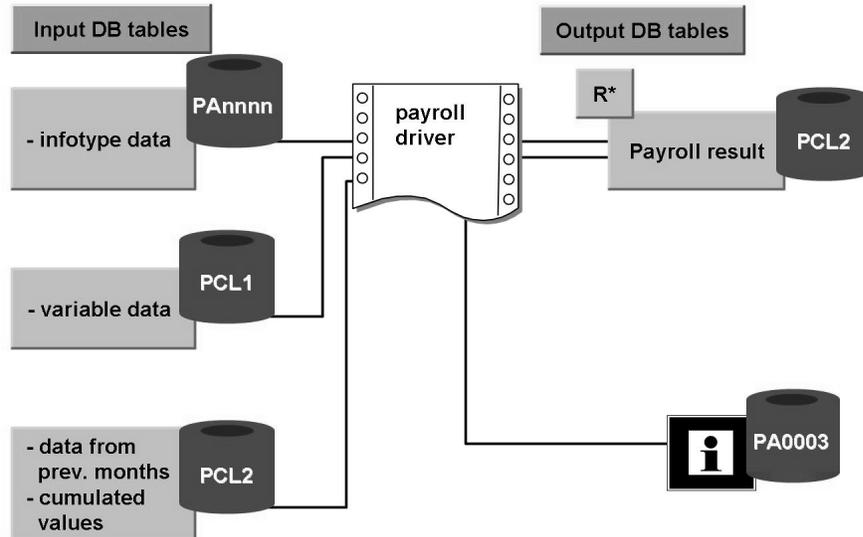


Figure 192: Input and Output Database Tables

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, and matchcode W.

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 the table, T500L.

The name of the cluster for the country-specific payroll results is stored in the table, T500L.



Cluster display: technical view



Figure 193: Printing Payroll Results

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

The cluster, RU, contains the results for the US payroll run, and the cluster, RD, contains the results of the payroll run for Germany.

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Index

A

ABAP commands, 133
ABAP List Viewer, 112
Active, 177
additional data pointer, 185
AND relationship, 203
Approved, 178
Authorization checks, 92

B

BAPI_EMPLOYEE_DE-
QUEUE, 89
BAPI_EMPLOYEE_EN-
QUEUE, 89
BEGDA, 182
Branch filter, 204

C

Concurrent Employment, 47
Contraction, 74
Customer-specific information,
225
Customer-specific infotypes,
228
Customizing, 173

D

Data and person selection
period, 38
Data model, 173
Data selection period, 38
Data structure, 3
Database tables, 131
DIALOG_MODE, 90
Display depth field, 203
Dynamic Actions table, 271

E

ENDDA, 182
Enhanced standard infotypes,
227
Enhancement concept, 231
Evaluation path, 201
Evaluation path field, 202
Evaluations, 67

F

Free delimitation, 52
Function module, 87

G

GET PAYROLL event, 153

H

HR Subroutines table, 271
HR_INFOTYPE_OPERA-
TION, 91
HRP1001, 181
HRPADnn table, 187
HRPnnnn table, 187

I

IMPORT statement, 140
Include screen, 232
INITIALIZATION event, 40
ISTAT, 182

J

Job, 175
Join, 67

K

Key fields, 6

L

Language-specific infotypes, 183
 Logical structure Pnnnn, 10
 logical structures, 15

M

Macro modules, 82
 MANDT, 182
 Modification group, 239–240
 Modification group 1, 240
 MOLGA parameter, 149

N

NSELC indicator, 237

O

Object filter, 204
 Object Infotype, 177
 Object-oriented design, 173
 OBJID, 182
 OR relationship, 203
 Organizational Management, 173
 OTYPE, 182

P

PADnn structure, 187
 PAKEY, 8
 Payroll driver program, 148
 Payroll result, 148
 Payroll results, 147
 PCLn database tables, 134
 Person selection, 38
 Person selection period, 38
 Personnel Administration infotype, 233
 Personnel Planning Infotypes, 251
 Planned, 178
 PLOGI table, 187
 PLVAR, 182
 Pnnnn logical structure, 187
 Pnnnn_VALID, 68

PNP, 29

Position, 175
 PP Infotype Enhancement, 251
 Primary key, 6
 PROJECTION, 74
 PROVIDE loop, 67
 PROVIDE statement, 74
 PSKEY, 10
 PSnnnn, 10
 PTnnnn structure, 187

Q

Qnnnn, 11

R

READ ONLY INTERNATIONAL parameter, 151
 Recursion check, 203
 Rejected, 178
 Relationships, 173–174
 Repetitive Structures, 107
 Report categories, 51
 Report class, 51
 Result Status, 154
 RH-CONDITION-LINE macro, 212
 RH-GET-TBDAT macro, 211
 RH_READ_INFITY function module, 213
 RH-SET-INDEX-INFITY macro, 212
 RH-SET-INDEX-INFITY-CONDITION macro, 212
 RP-IMP-C1-TX, 138
 RP_PROVIDE_FROM_FRST macro, 83
 RP_PROVIDE_FROM_LAST macro, 83

S

SAPDBPNP, 82
 Selection screen, 38
 SEQNR, 182

Sequential evaluation, 199
Sort function, 41
Sort sequence, 41
Standard infotypes, 226
Status indicator, 148
Status overlap checkbox, 202
Status vector field, 202
Structural evaluation, 200
Submitted, 178
SUBTY, 182
Subtypes, 4, 17

T

Table infotypes, 185
table pointer, 186
Technical depth, 203
Time constraint, 5
Time Constraint, 4
Transparent table, 6

V

Variable key, 242
VARYF, 182

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Feedback

SAP AG has made every effort in the preparation of this course to ensure the accuracy and completeness of the materials. If you have any corrections or suggestions for improvement, please record them in the appropriate place in the course evaluation.