Pension Fund CH: Reference Guide

HELP.PYCHNTPF

Release 4.6C

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
➡	Note
🧭	Recommendation
<SYN>	Syntax
💡	Tip

Contents

Pension Fund CH: Reference Guide

Concept of the SAP PF

Purpose

The SAP PF module supports the management of insured persons within a pension fund through the following functions:

- Calculation of PF-relevant pay and contribution data
- Payroll interface
- Calculation of benefits
- Reporting

Implementation Considerations

The necessary steps to set up the PF module can be read in the corresponding Implementation Guide (IMG).

Integration

Technically, the PF module is part of HR and contains the following additional functions:

- PF-specific infotypes
- Processing control (see below)
- PF workflows
- Account management
- Diverse reports
- User-interface
- Diverse tools

Background

For most Swiss employees, the company pension is the main element of their retirement (the 3-pillar model: AHV, PF, private pension): The resulting pension is usually 60 - 70% of the last gross pay, while the AHV pension can be seen as the base pension.

For historical reasons, there are a large number of PF regulations. The first statutory rule was introduced in 1985 with the LOB. This rule only provided a rough outline, and guaranteed certain minimum demands (for example, estimates for registered pension institutions). The existing regulations can be roughly divided into two types: Benefit Plan and Contribution Plan. In the first case, benefits are determined using individual data records. In the second case, (and savings funds) an age-related credit is carried out (the LOB estimate calculation follows the savings fund model). In practice, you often come across a mixture of these two, in that savings bank benefits are (partly) calculated according to the benefit plan. For standard software, this presents a major problem in meeting the different demands and needs of the customer, which requires a great deal of flexibility.

Pension funds differ, in that some are “registered”, which means that they come under the LOB. In this case, you must carry out a LOB estimate. An employer must be registered with this type of pension fund. Additional, voluntary benefits can be paid through a non-registered fund (*complementary* funds, cadre funds). Registered pension institutions with voluntary benefits are known as *comprehensive*. Most funds are comprehensive.

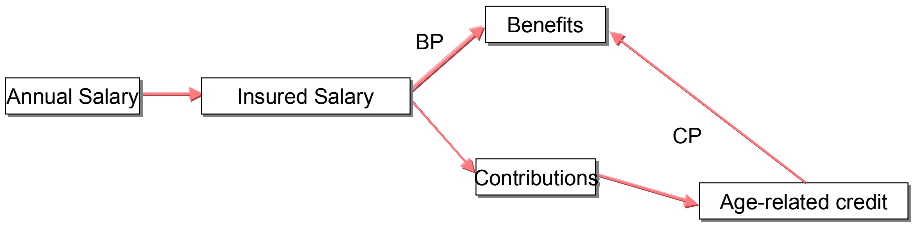
Another special feature is that a company pension fund is legally independent, which makes it difficult for a company to gain access to the funds. This division enables different departments to manage personnel and pension funds.

The pension fund normally uses annual values for calculation purposes. Short-term income changes are not normally taken into account. At the start of the year, you must determine new contributions based on the modified basic data (salaries, age, and coordination deduction).

For many funds, you must also consider individual agreements such as an employee’s increased voluntary contributions, in addition to the regulatory requirements.

PF Module Functions

The following diagram shows the most important PF administration terms:



- The annual salary is normally 13 months' pay. You must also take into account wage elements from the previous year. Mean value formations can also occur. Unfortunately, the purpose of this value is not clearly defined in the PF regulations.

- The *insured salary* is the annual salary minus the coordination deduction and the limit. Use this to coordinate with the two other retirement pension pillars. The calculations are highly fund-dependent.

- Typical pension fund *benefits* are: Retirement pension, Invalidity pension, Widow's pension, Orphan's pension, Pensioned/Child pension, Death-related equity, Death benefit, and AHV interim payments.

- The benefits are financed by the *contributions*. *Risk contributions* are generally displayed separately (for 18-24 year olds, only the risk benefits are insured). *Savings contributions* must be paid for savings funds. These go into the age-related credit. For benefit plan funds, you usually differentiate between *normal contributions* to finance the current insured benefits and *exceptional contributions* for wage increases.

- The *age-related credit* (and savings credit) are only shown for contribution-plan funds and provide the basis for benefit calculation. The LOB age-related credit is a special case. This is also required for the administrative PF estimate in the benefit-plan funds.

Calculations with the *BP* arrow correspond to the benefit plan model. Those with the *CP* arrow correspond to the contribution plan model. In the following, a pension fund is referred to as contribution plan fund, if its rule provides the formation of an age-related credit (apart from LOB age-related credit) to calculate benefits.

To effectively support PF administration calculations, the following are required:

- A possibility of calculating and deducting contributions. It must be possible to ignore short-term salary changes or certain salary elements.

- A possibility to calculate benefits; account management is especially important here.

Implementation

The PF module is highly flexible. You can modify the form and operation of its programs in Customizing, without altering ABAP programs. The range of functions is not pre-set. You can expand it as needed. You can set up SAP PF, as if it was specially developed for your regulations.

You can convert a PF regulation for a desired functionality within a “processing schedule”. Here, you must define various calculations – and general procedures – for the pertinent fund.

You can change the type of report printout by modifying the SAP*script* forms.

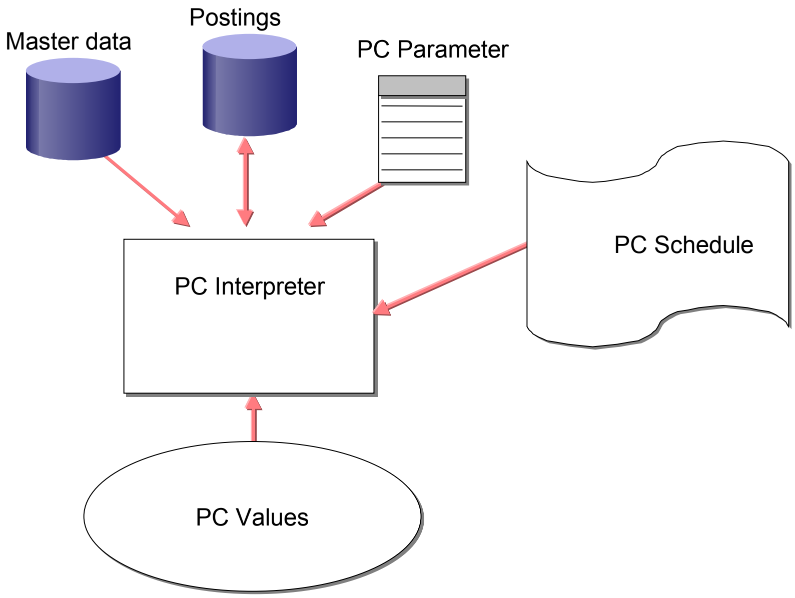
You can model the processes and communication (across departments) with workflow-management.

SAP*script* and Workflow Management are established SAP basis technologies, whereas the processing control concept was developed specifically for pension funds. The SAP system does not support all possible PF regulations- Instead; it provides examples that you can modify.

Processing Control: Overview

Use

The following diagram outlines the basics of processing control:



Integration

When you are working with the processing schedule, you can access HR master data and the PF accounts, and the *PC Parameter.* In this way, you can edit the table values (for example, contribution records, current value factors).

Features

The PC-Interpreter is implemented by a number a subprograms. It is usually started with a function module. Here, you must specify the *PC-Activity,* the *processing schedule,* and the *processing context*.

Value types

Definition

Values are the core of processing control (PC). Values are identified and described through *value types*.

Use

A processing control value has one of the following data types:

- Whole number
- Number to 2 decimal places
- Number to 4 decimal places
- Character string
- Date
- Value type
- undefined

This data type is assigned a current existing value through the corresponding value type. You can control possible entries, entry checks and processing using the data type.

You must calculate the total of two values assigned to the value types "XXXX" and "YYYY". If XXXX and YYYY are both datatypes with *numbers to two decimal places*, the result will also be to two decimal places. A corresponding addition is carried out. If XXXX is from the data type *Date,* and YYYY from data type *Whole number*, the result is the total of the *Date* datatype, (where the second value is the number of days to be added to the date of the first value), and which forms the relevant result.

PC Interpreter/PC Schedules

Definition

The *PC-Interpreter* is the core of processing control, which processes values by processing PC schedules.

A processing schedule is a SAP - or customer - object. Apart from the current schedule, there are further schedule-dependent settings (see below for details):

- Activity assignments
- Specific value type settings

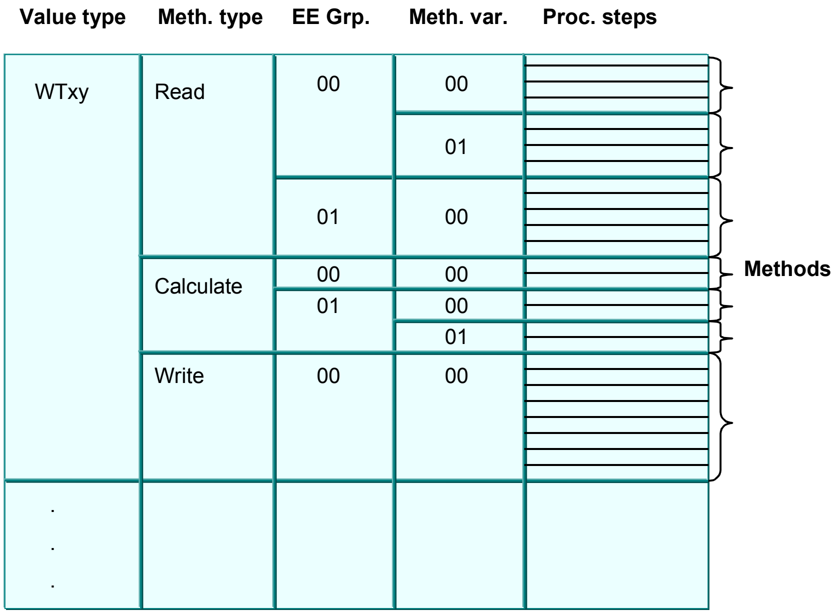
Use

A processing schedule consists of a number of *Methods*, assigned to value types. They can be divided according to *Method type* (for example, read, calculate), *Employee grouping* (for example, all insured persons, sales executives) and *Method variant* (that is, validity period).

Within a processing schedule, methods are addressed in the form of *Value type-Method type* (→ *sub method call up)*. Additional key terms are determined at run time.

Structure

The following diagram shows the structure of a processing schedule:



A method consists of a series of *processing steps*. The following are defined in the processing schedule for each processing step:

- An operation to be executed (reference to ABAP-subprogram)

- An amount (max. 10) of operation parameters
- A line type (defines the processing type of the operation result value)
- The next processing step (dependent upon the operation return code).

PC Activity

Definition

PC Parameters are specially determined values within the processing control. They are obtained by selecting the relevant table lines from table values.

Use

A PC parameter is configured with reference to gender-dependence, and two other key terms. In effect, you have a validity period.

PC parameters are grouped using *parameter areas*, according to their configuration (key fields, gender dependence) and formation.

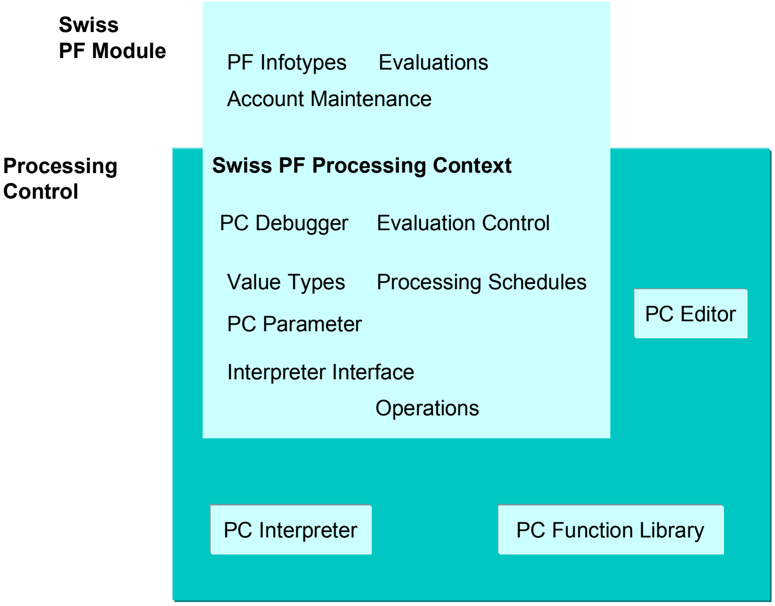
In the PF module, PC parameters are used for contribution records or present value factors.

Processing Contexts

Definition

Processing control is a tool used in PF. It is feasible that the concept and certain elements of processing control will be used in other modules in the future. To enable this to happen, the term processing context was added. The processing context is formed from the country code and an indicator for the sub-application. A range of processing control objects is dependent on the respective processing context.

The following diagram shows the terms and their relationship to one another:



Use

Note that the actual core elements of processing control “know” nothing about the respective problem area. Therefore, you must always specify the processing context for the programs and the function modules of the processing control, to which the respective activity relates.

In the PC settings, you cannot directly specify the fund, as this is a special entity of the *Swiss pension fund* processing context that is not recognized in other processing contexts. Instead, other terms are added, as necessary (such as parameter range determination grouping), to which the terms of the current problem area (for example, fund) are assigned.

PC Activities

Definition

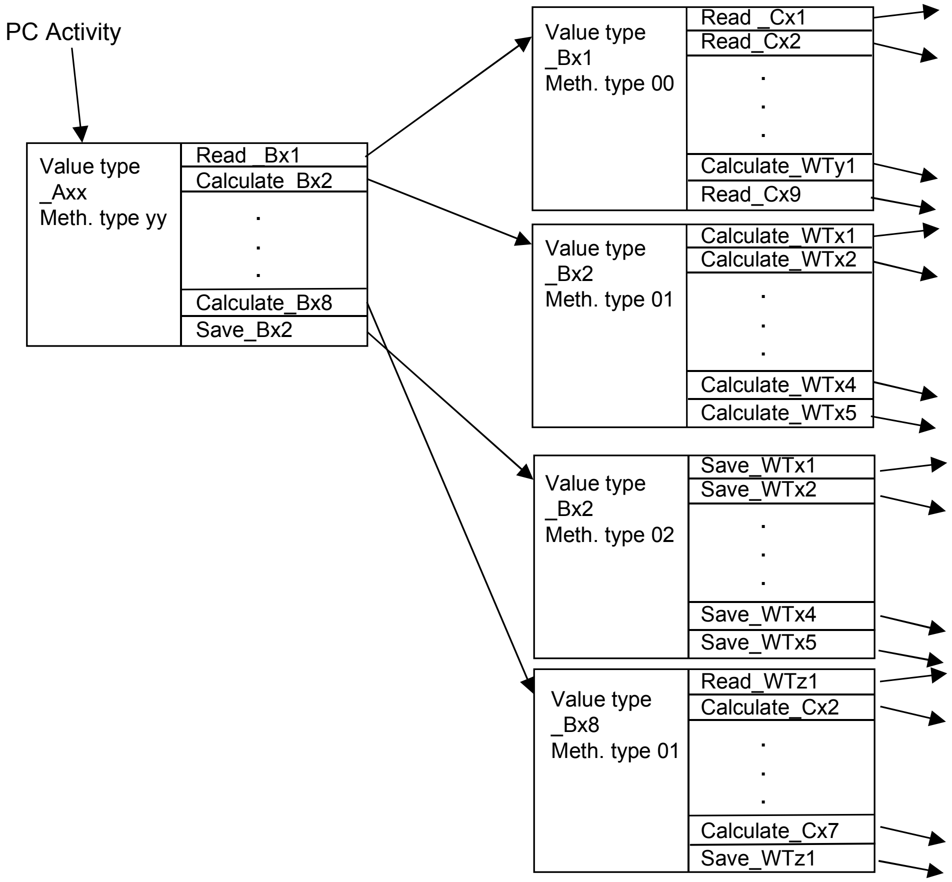
The PC Interpreter is called up using a function module modified for the respective processing context. This call-up corresponds to the PF module programs.

How can you be certain that you are using the correct methods, in the correct order, in the processing schedule?

As previously mentioned, you can call up sub-methods. For a special processing, you need only define a method that calls up the required methods in order, as sub-methods. Assign this method to the Interpreter as the “entry point” in the processing schedule, in the form of a *Value type-Method type*. Assign a *PC activity* to this method as an abbreviation. The PC activity is then assigned to the Interpreter upon call-up.

Use

The processing of submethods is shown in the following diagram: Only the value type and the method type are relevant for the call-up structure; employee grouping and method variants are left out to simplify the diagram:



Funds

Definition

In the PF module, a *fund* is a pension institution with its own regulations. It is used to differentiate between several (possible) employee insurance relationships.

Use

Carry out the following fund-dependent settings:

- The configuration of master data; the fund is used as a subtype
- The assignment of a processing schedule and *Parameter grouping*
- The payroll interface
- PF recalculations (user interface)
- PF functions (user interface)

Master Data

The infotypes *Basic Data PF* (0278) and *Individual Values PF* (0279) are part of PF. The fund is a subtype.

Both can be configured in the usual variety of ways.

The SAP standard system has a pre-set time constraint of “2”. That is, only one infotype record can exist at any one time for each subtype. Gaps are allowed.

The usual setting must exist for these infotypes, in particular in T777D, T582A, T588M, T588I and T588J. If necessary, you must make adjustments in client 000.

Basic Data PF Infotype (0278)

Definition

This infotype contains the insurance data that rarely changes, such as entry date or degree of invalidity.

The infotype has up to eight employee attributes, which you can configure along with the *Fiscal Data CH* (0038) and *Additional Organizational Assignment CH* (0039) infotypes.

Individual Values PF Infotype (0279)

Definition

This infotype contains wage and contribution data; that is, data that changes on a regular basis.

Structure

The infotype has 20 fields, which can be assigned value types using the view V_T5CP3. The assignment of value types takes place independent of fund and employee grouping. In this way, you can modify the form of the infotype to suit your requirements. The employee grouping is determined using the CHPKP feature. In this way, you can make different field assignments for active employees and retirees. At the same time, you can determine the outline text independently of fund and employee grouping.

Each logical field assigned in this way consists of two screen fields, where the right screen is purely a display field. This helps to distinguish between calculated and manually overridden values. The display field contains a calculated value if and when:

1. A calculated and a manual value exist for this field, and
2. These values are different.

Calculated values go into the infotype through workflow or Batch Input.

You can assign the individual infotype fields –- dependent on fund – to processing activities, so that online-calculations are possible, according to the processing schedule (for example, changing the insured salary → retroactive calculation of contributions). For performance reasons, these calculations cannot be done with Batch Input.

The *Individual Values PF* infotype (0279) contains retroactive accounting-relevant fields (contributions). In the standard system, the infotype 0279 is defined as wholly retroactive accounting-relevant. This can result in unnecessary retroactive accounting, if certain non-contribution fields are changed. To avoid this, define the retroactive accounting trigger as *field-dependent according to T588G* (Table T588G, Field *Retroactive Accounting Relevance for Payroll* = 'T'). You can use CHPKP directly for the *Variable Key* in T588G, to take into account the different field assignments for the various employee groupings.

Pensions do not have to trigger a retroactive accounting run, as pensions are also entered in the *Basic Pay* (0008) infotype: The infotype is also retroactive accounting-relevant.

You can expand the list display for the *Individual Values PF* (0279) infotype through Customizing. For more information, see the Implementation Guide (IMG) under *Master data –> Individual Values PF (279) –> Define column headers for list display/Assign fields for list display*.

Create Infotype 0279 During Dynamic Action

To create *Individual Values PF* infotype (0279) directly in the entry action, use the PF new calculations in the user interface.

See also:

Using External Programs [Page 92]

Account Management

PF Account Name

PF accounts are to be seen as technical accounting (not endowment accounting). They form the basis for PF benefit calculations and statistics.

A PF *account* can be seen as a container for PF-relevant *postings*. An account is identified by an *account number*. This account number is assigned to an insured person – time-dependently. (Data modeling also allows you to assign an account to several personnel numbers; however, this is currently not used.)

A posting contains the following information:

- Posting date
- Value date
- Value type
- Value
- Origin
- User name of the last person to change the data

Go to posting maintenance using the transaction **PACO**.

The symbol in front of the posting specifies whether the account receives a credit (for example, a positive symbol (+) for a contribution, or equity payment) or a debit (for example, a negative symbol (-) for a pension payment, or advance).

Three pieces of data are assigned to an account:

- **Start/End date**: This specifies the valid time period when you can enter postings; the posting date plays a definitive role here.
- **Lock date**: When creating a posting, the value date cannot be prior to the lock date.

Specify the validity period when you create the account. The start date should be, at the earliest, the insured person's start date. The end date is usually 31.12.9999. Checking the lock date prevents you from creating postings for the time period prior to the last account closure. After the account closure (on 31.12.YYYY), the lock date is changed to the following day (01.01.[YYYY+1]).

Changes to start, end or lock dates are only possible with a special authorization.

The following are also assigned to an account:

- An **account type**. Account types are used to separate the accounts according to certain criteria. For example, you can set up an Invalidity account, where a partial invalidity is assigned, and into which invalidity payments are made, and into which contributions are paid, depending on the degree of invalidity.
- A **currency**. All posted amounts in the account must be in this currency. In this way, you can display totals regarding particular postings. Use the currency for the country grouping '02'. It is determined as follows: A personnel area with the country code '02' is read from table T500P, according to the database. Normally, this is the first item in the canonical sorting

sequence. The currency is then the national currency (according to T500C) for the company code.

- Certain internally formed information, such as **date** and **user name** of the last person to change the data, and an indicator whether open postings exist.
- **Authorization group**.
  This authorization group is set when the account is created and is used to control access to the account, together with the authorization object P_CH_PK.

At this time, only one account can exist for each personnel number for a fund, per account type.

Account Numbers

Account numbers are managed using the number range object *HRCHPKONTO*. The number range is determined when you create an account, with the feature CHPKN.

In connection with authorization concept for PF accounts, you can control the access authorization using specific data for the account holder. The authorization to create an account can only be controlled with the account number, as the authorization group is not yet determined.

Account Maintenance

Accounts

You can maintain an insured person's accounts with the transaction PACB. The following functions are available:

- Create an account
- List accounts, Go to postings maintenance
- Change validity period and lock date

Manual postings

You can maintain account postings with transaction PACO – usually started with the transaction PACB.

You can still use the previous posting maintenance PACE using the name *classic posting maintenance*.

Account views

With account views, you can summarize postings under business management criteria, so that when you select *Savings credit EE* in the list, it exactly displays the postings relating to the employee savings credit.

Also, you can assign PC activities to the account views, and display the results in the navigation window.

Set up account views in the PF IMG under *Accounts –> Settings for posting maintenance*.

Authorizations

Berechtigungsobjekt P_CH_PK [Page 32]

Berechtigung für Einstellungen [Page 33]

Use

Apart from the existing HR authorization concept, (for master data, for example), special authorizations are also required to access PF accounts. Therefore, a new authorization object **P_CH_PK** was added.

Integration

General authorization checks (for example, authorization for transaction call-ups, and master data maintenance) are configured and executed within the relevant system component.

Features

The authorization object P_CH_PK checks whether, and to what extent, a user can access account data.

Authorization Object P_CH_PK

Definition

The authorization object has the fields *Account number* (KONNR), *Authorization group* (AUTGR) and *Authorization level* (PKKLV).

The following authorization levels can be assigned:

Value	**Meaning**
-	No account access
R	Read authorization
W	Maintenance authorization
X	Advanced account authorization (Conversion of Validity period, lock date)

Use

This authorization object is evaluated

- in the account/posting maintenance transaction
- in the account access function modules
- in processing control.

Authorization for Settings

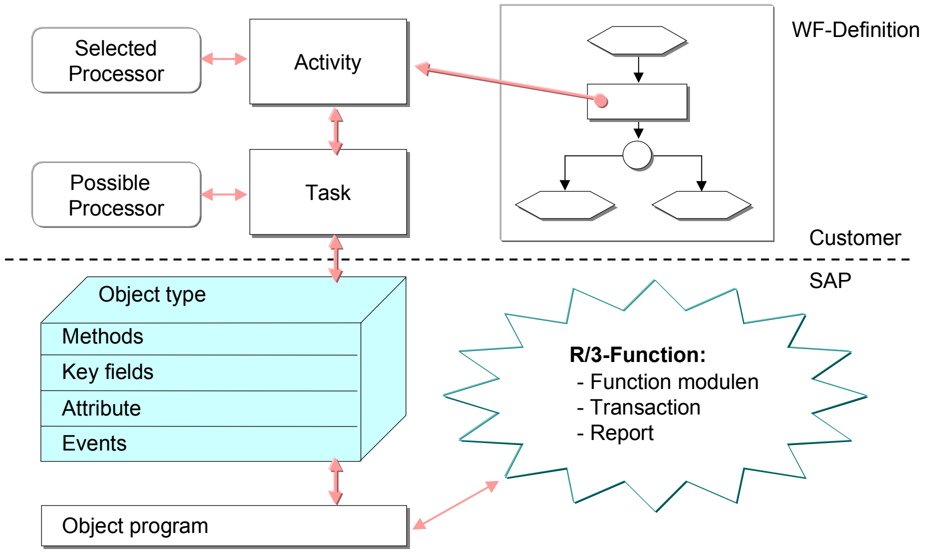
Definition

An authorization is required to maintain processing schedules. This is the authorization to carry out transaction **PAC0**. If you do not have this authorization, you cannot start the PC Editor.

Processes with Workflow Management

Use

The following diagram provides a technical overview of the basic terms of Workflow Management in the standard SAP system:



To carry out activities within Workflow, you require the relevant object type methods. These object types are summarized in the *Business Object Repository* (BOR) (Transaction SWO1).

Business Object Types in HR

Definition

In HR, each infotype can correspond to a business object type, which is a sub-object type of **EMPLOYEEIT** or **EMPLOYEESU**.

Use

The V_T588V view assigns infotypes to the corresponding business object types.

Each infotype module pool contains codes to create events for DB operations. This event-creation is individually controlled by the V_T588W view. (From Release 4.X : V_T779W/V_T779X. You can override the customer-specific SAP settings using V_T779X. You can find this setting in the IMG, under *Personnel Administration –> Tools –> Business Workflow Events –> Redefine Event Types for SAP Business Object Types.*)

There are two possibilities here:

- Event creation for database operations [Page 37]
- Event creation for function modules [Page 38]

Structure

The object types **EMPLOYEEIT** and **EMPLOYEESU** have the following interface:

Methods		
	CreateInteractive	Create
	DeleteInteractive	Delete
	Display	Display
	ChangeInteractive	Change
	ExistenceCheck	Check existence
	Find	Find
	FindPredecessor	Find predecessor
	FindSuccessor	Find successor
	FindWithoutDialog	Find without dialog
	CreateSuccessorInteractive	Create successor
Key fields		
	EmployeeNumber	Personnel number
	Subtype	Subtype
	ObjectID	Object identification
	LockIndicator	Lock indicator

Business Object Types in HR

	ValidityEnd	Validity end
	ValidityBegin	Validity start
	RecordNumber	Record number
Attribute		
	ObjectType	Object type
	InfotypeNumber	Infotype number
Events		
	CREATED	Created
	DELETED	Deleted

Event Creation for Database Operations

Each DB operation is assigned to an object type event. Possible DB operations are: Insert (INS), Update (UPD), Delete (DEL), Request (REQ), Approve (APP), Approve Cancel (APC) and Request Cancel (RQC).

Note that simple operations – on the business management level – such as delimiting an infotype record, affect several DB operations. If a change in the key for an infotype record occurs, (for example, a new validity end), the previous record is deleted (DEL) and then recreated with the new key (INS).

Event Creation for Function Modules

In many cases, the name of the event creation does not match the name of the object type events when using DB operations. This is because they are dependent on the technical implementation of business management steps.

Using the function module HR_EVENT_RULES_PAnnnn for the infotype nnnn and DB operations assigned using T779W/X can solve this problem. The function module receives data through its entry parameters that enable you to closely control the event creation.

Business Object Types for HR

Definition

There are two object types for Swiss PF that correspond to the existing infotypes:

See also:

- PFBasicCH [Page 40]
- PFValsCH [Page 42]

PFBasicCH

This is the corresponding object type for the *Basic Data PF* (0278) infotype. The supertype EMPLOYEESU interface is expanded as follows:

Methods		
	CheckAttribute	Test PF Attribute
	CreateAccount	Create PF Account
	ProposedValuesForInfty	Determine Default Values
	CreateInftyWithProposal	Create Infotype with Default Values
	ListEntries	List Account Postings
	CreateEntryWithoutDialog	Create Posting (Background)
	CreateEntry	Create Posting
	ListAccounts	Display Employee Accounts
	GetPFBasicDataList	Create List of PFBasicCH Objects
	PrintInsuranceReport	Print Insurance Statement
	CheckValtyProperty	Check Value Type Property
	AdditionalEntries	Enter Additional Postings (per Dialog Window)
	IntAge	Calculate Age in Whole Years
Attribute		
	Aggregate	Chronological Employee Characteristics
	Description	Subtype Name
	ExternID	External Indicator
	ReservationDate	Reserve until Date
	InvalDegree	Degree of Invalidity
	DateOfEntry	Entry date
Events		
	EntryCompleted	Entry Completed
	changed	Changed
	validitychanged	Validity Period Changed
	EntryStarted	Entry Workflow Started (technical)
	ValuesCreated	Individual Values Created
	SalaryForPFchanged	Salary for PF changed

	EntryCreated	Posting Created in Account

PFValsCH

This is the corresponding object type for the *Individual Values PF* (0279) infotype. The supertype EMPLOYEESU interface is expanded as follows:

Methods		
	PC Activity	Execute PC activity
	SuccCreateWithVals	Create successor with default values
	CreateWithVals	Create with default values
	GetValueFromTab	Get value from table
	WriteValueToTab	Write value to table
Attributes		
	Aggregate	Employee time specifications
	Description	Subtype name
	PlanID	Processing schedule for fund
	Currency	Currency
Events		
	changed	Changed
	validitychanged	Validity changed

PF Workflow Structure

Standard workflows are structured as follows:

- The actual processes are fund-dependent.
- They are integrated as sub-workflows in one distribution-workflow.
- If necessary, there is a further Toplevel-Workflow.

See also:

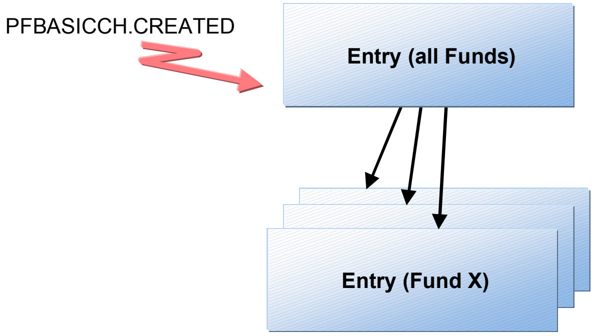
Entry [Page 44]

Changes to Object Type BASICPAY (Infotype 0008) [Page 45]

Delimit Individual Values PF Infotype (0279) [Page 46]

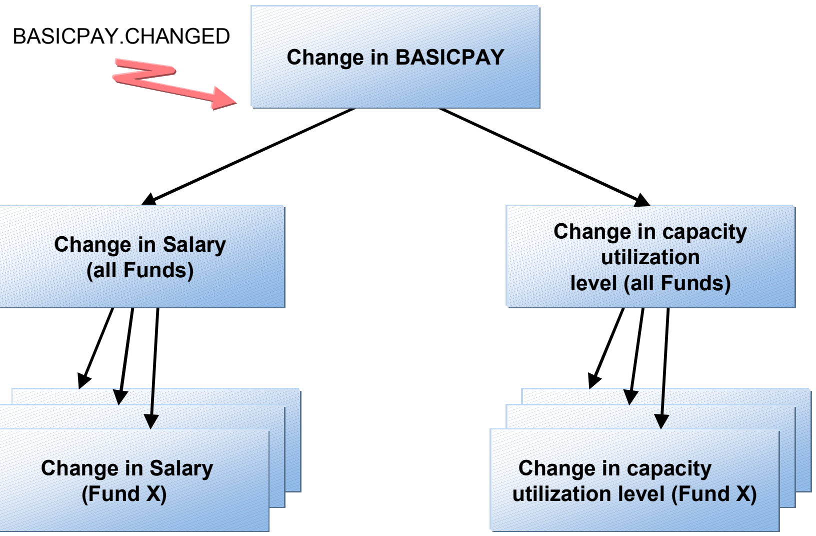
Processing of Payments [Page 47]

Entry



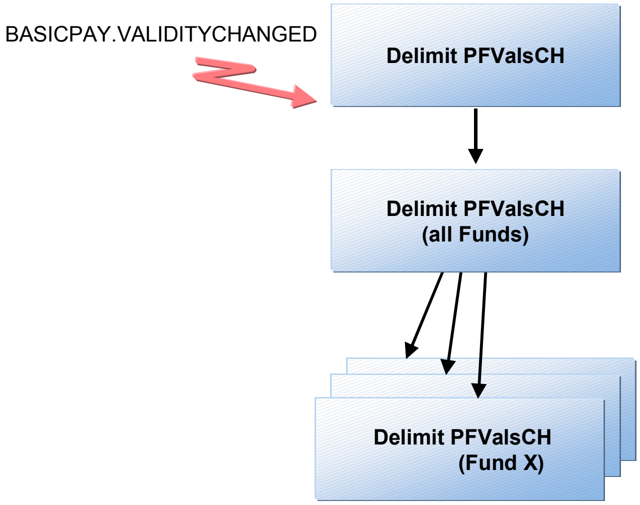
Changes to Object Type *BASICPAY* (Infotype 0008)

The Toplevel-Workflow *Change BASICPAY* examines the type of change and, if necessary, calls up the corresponding distribution-workflow for all of the employee's funds.


BASICPAY.CHANGED
Change in BASICPAY
Change in Salary (all Funds)
Change in capacity utilization level (all Funds)
Change in Salary (Fund X)
Change in capacity utilization level (Fund X)

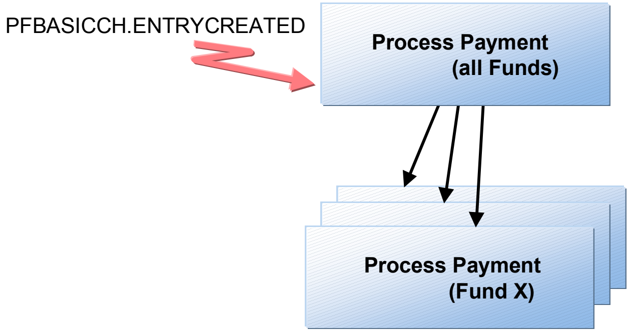
Delimit *Individual Values PF Infotype* (0279)

First, in Toplevel-Workflow, the list of funds list is determined, with which the employee is insured. A distribution-workflow is started for each of these funds, which calls up fund-dependent workflows with the actual processing.



Processing of Payments

The distribution workflow is directly started, which then starts the workflow for the respective fund in a multiple selection.



Description of PF Workflows

Customizing von Workflows [Page 54]

In principle, a workflow can exist for each fund. In the standard system, workflows for contribution plans and benefit plans differ.

Entry

Entry consists of the following steps:

- Creating an account
- Calculating wage/contribution data
- Calculating labor mobility benefit (LMB), if necessary
- BP: Recalculating wage/contribution data after posting the labor mobility benefit insurance statement

If necessary, modify the default value type for posting the labor mobility benefit

Change in salary

This process has the following steps:

- Recalculating wage/contribution data
- BP: Entering a retroactive payment for the indemnity capital
- Insurance statement

If necessary, modify

- the wage type to create the retroactive payment (infotype 0014)

Change in the Capacity Utilization Level

This process has the following steps:

- Recalculating wage/contribution data
- Insurance statement

Delimit *Individual Values PF* Infotype (0279)

As the infotypes 0008 and 0279 are usually held at the same time, you must notify any postponement of the start or end date in the *Basic Pay* (0008) infotype to the PF. This takes place with the event BASICPAY.VALIDITYCHANGED. A workitem is created for the record with the changed start date. You can use this workitem to change the corresponding record in infotype 0279.

Processing of Payments

For incoming payments, (for example, LMB), you must recalculate the pension rate for the benefit plan.

If this results in a change in benefits, you must create an insurance statement (for the benefit plan also).

Customizing of Workflows

Purpose

The workflows set up in the standard system are defined as models and are therefore SAP objects. Therefore, it is recommended that you copy these to customer objects (workflow tasks), as models in future SAP releases may be altered. You can, to a certain extent, control the operation of workflows in the standard system by defining the initial values in the respective Toplevel workflow. Specify a PC administrator in the field *US<User name>*, so that errors during the background step *PC Activity* can be reported by mail. Therefore, you must copy the Toplevel workflows.

The workflow activities do not contain user assignments. This is done at the individual task level.

Set parameters for the workflow *Change in BASICPAY*

The workflow activities contain no processor-assignment. Assign processors as a single-step task.

Procedure

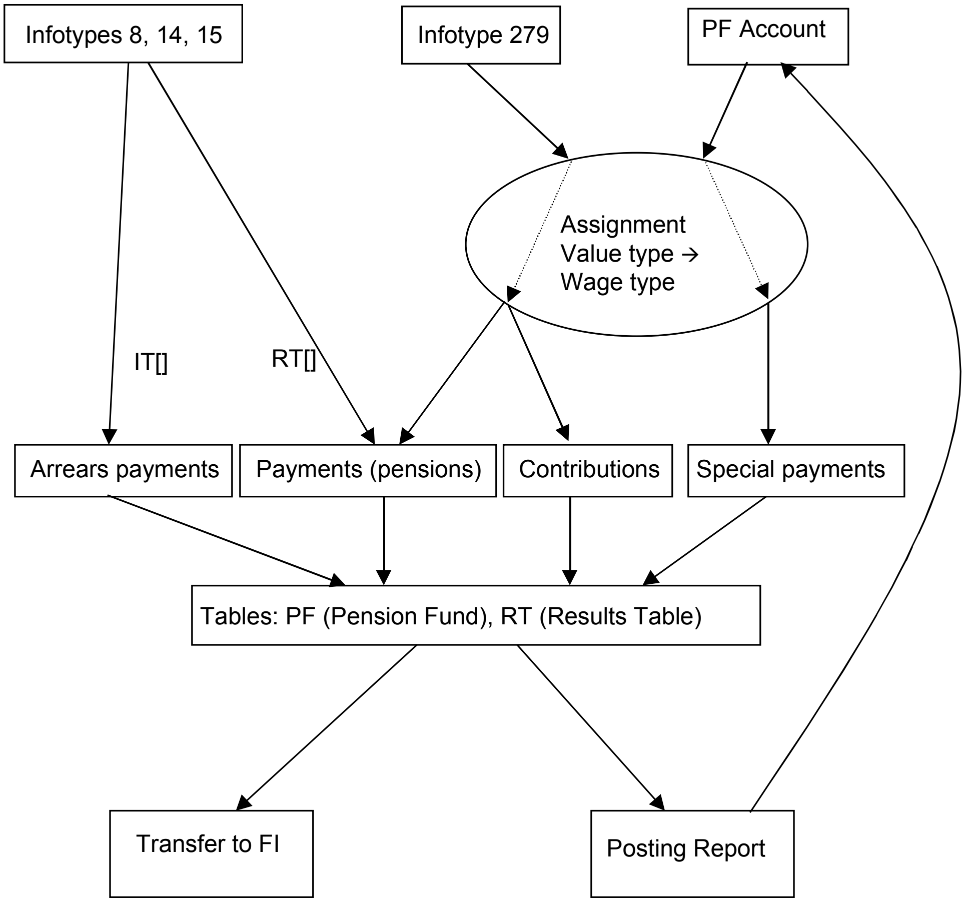
1. Maintain the **organizational plan** with a PF department (Workflow-development environment: *Definition tools*→ *Organizational plan* → *Create*/*Change*). If you need to implement retroactive payments (BP) in infotype 0014, using workflow, a personnel department will be necessary.
2. Create a PF department **position** (for example, PF clerk).
3. Maintain the **task profile** for the position: Select the tasks with *HRCHPK**, select all and assign them. If necessary, you can transfer the task *Enter retroactive indemnity-capital payment (deduction)* to the personnel department.
4. Assign **system users** to the positions.

   ⚠ Note that user-assignment of single-step tasks and event-linkage is *client-dependent*.

Payroll Integration

Use

The following diagram shows a summary of payroll integration:



Features

Payroll integration has the following functions:

- Deducting contributions
- Crediting pension payments
- Transferring manual postings to financial accounting

Implementation

The following aspects are essential for the integration of PF in payroll:

- The assignment of wage types to value types (T5CPA),
- Including payroll periods and years in the PF postings

Only the PF value types that have been assigned a wage type using V_T5CPA are processed in payroll. There are three different types:

Contributions	The values for the T5CPA value types are taken from the *Individual Values PF* (0279) infotype, where the subtype corresponds to the specified fund. The infotype must have entry fields relating to the existing fund that are assigned to value types. The wage type determined for a contribution value type using T5CPA goes into payroll as a deduction wage type. A corresponding entry is written in the PF payroll table.
Pensions	Pensions are processed in the same way as contributions. However, the corresponding values must be present in the payroll infotypes (for example, infotype 0008 or 0014), with the wage type assigned via the view V_T5CPA. These values must also have been transferred to the table RT[], prior to calling up the CHPK function, so that a credit results. In this case, a corresponding entry is also written in the PF payroll table. For wage type category R, each of these values must be present in the current infotype 0279, with the same amount. For wage type category P, the existence of infotype 0279 is not checked.
Arrears payments	Arrears payments (exceptional contributions) aid in the processing of arrears payments for wage increases in the benefit plan. When you call up the CHPF function, the corresponding values must already be present as wage types in the table IT[] (usually from infotype 0014). In this case, the contents of infotype 0279 are irrelevant.
Special payments	This value type category is used to transfer postings from PF accounts to financial accounting. The postings are transferred to payroll, provided that payroll is run for the period in question. In this case, the value of the posting is cumulated in the wage type assigned per T5CPA. This makes a transfer to financial accounting possible. For postings processed in this way, corresponding entries are then written in the PF payroll table.

If the account is already locked, (⇔ End of the payroll period < Lock date), the payroll run in function CHPK stops, as PF postings have to be changed.

The report RPUPENC0, which is started immediately after the payroll run, updates the PF accounts by evaluating the payroll table PF.

Contributions	If postings with the corresponding value type for the payroll period already exist, they are cancelled, if the total of these postings is not zero or different from the new posting value. If a posting is required, it is created.
Pensions	As contributions, except that pensions are posted as a negative value.
Arrears payments	As contributions.
Special payments	These postings are closed according to their availability in the PF payroll table.

Processing Classes and Evaluation Classes of PF Wage Types:

Process. Class	Contributions	Pensions	Arrears payments	Special payments
1		1		
3		0	0	
4			9*	
6	0	0	0	0
10		A	0	
20	1	3	1	1
30	0	0	0	0
31	0	0	0	1
32	2 / E (EE/ER)	A	2 / E (EE/ER)	2
61	3 / 4 (EE/ER)	5	3 / 4 (EE/ER)	2

- *If the respective wage type is defined as a deduction wage type, otherwise 0.*

Eval. Class	Contributions	Pensions	Arrears payments	Special payments
1	20 / CB (EE/ER)	FA	20 / CB (EE/ER)	20
2	27 / 28 (EE/ER)	20	27 / 28 (EE/ER)	
3	03 / 04 (EE/ER)	01	03 / 04 (EE/ER)	03
4	20 (ER)		20 (ER)	
5	1D (EE)	RT	1D (EE)	3M
6				

The following table summarizes the wage type characteristics:

	PF Posting	Source	Infotype 0279
Contributions (C)	created positively	Infotype 0279	required
Pensions (P)	created negatively	Wage type/Infotype 0279	required
Pensions from payroll (P)	created negatively	Wage type	–
Arrears Payments (AP)	created positively	Wage type	–
Special Payments (SP)	closed	PF Account	–

Transfer

All wage types processed in a payroll run can be transferred to financial accounting. With the exception of PF contributions, this is also possible for certain manual postings, using the mechanism sketched out above.

Note that endowment accounting is generally in a different posting area from payroll.

Updating the Individual Accounts

Use

To update the individual accounts using payroll results, run the report RPUPENC0.

Features

The report has the following functions:

- Posting of contributions and pensions.
- *Closing* of manual postings that have already been processed in payroll.

Payroll Schema without Salary

Use the payroll schema CH10 to process the contributions for externally insured persons. Generally, this is set up so that you do not need to make any significant changes to Customizing. In particular, you can use the same wage types for PF contributions as for schema CH00.

Schema CH10 requires the following master data:

Actions (0000)
Organizational Assignment (0001)
Personal Data (0002)
Addresses (0006)
Basic Pay (0008)
Social Insurance CH (0036)
Fiscal Data CH (0038)
Individual Values PF (0279)

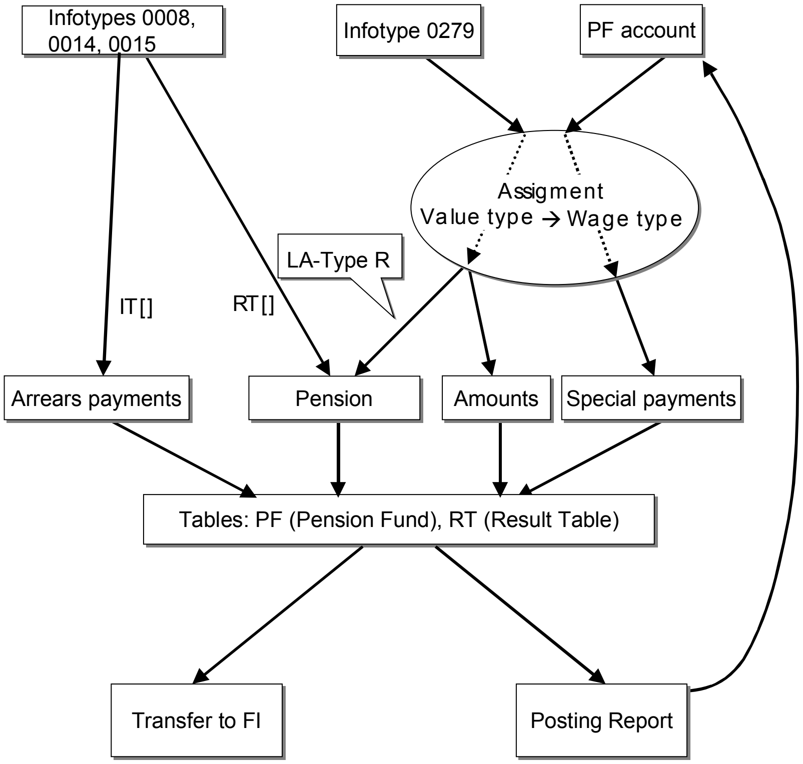
Without all of this master data, you cannot run payroll for this person with the standard schema. You also need the *Basic Data PF* (0278) infotype to administer the PF.

You also have a mechanism to prevent the processing of regular employees using schema CH10: to do this, you must replace the personnel calculation rule CH1C in schema CH10 with a specifically modified copy.

For more information, see the schema CH10 and personnel calculation rule CH1C documentation.

Summary of Payroll Integration

The following diagram shows a summary of payroll integration:



PC Universal Evaluations

VS-Schemata [Page 66]

Use

Universal evaluations make up a significant part of the PF module. You can implement them using a few reports, and they enable you to carry out wide-ranging modifications and enhancements, without changing the program.

An *evaluation type* is assigned to each of the various reports that implement the universal evaluation. In this way, you can maintain report-specific settings.

Evaluation type	Report	Description
04	RPLPKBC0	Applicant evaluation SAP*script*
05	RPLPKSC0	Universal evaluation for employee data

The individual reports „know“ their evaluation types (defined for each as a constant). In this way, the correct types are selected for the report-dependent settings.

Features

The following functions are supported:

- Choosing the desired process on the selection screen
  Display with SAP*script*
- Additional entry parameter in selection screen
  for data not in the system (for example, applicants, simulations)
- Creation of Batch-Input-Sessions in line with Customizing
  (Evaluation type 5)
- Export to Word/Excel
  (Evaluation type 5)
- Definition of group formations (4 levels) with subtotals,
  In addition, 1 grand total, (Evaluation type 5)

The basic concept behind universal evaluations is to separate the creation and processing of data from the display. This is possible, because the calculated data is stored in an internal table. You can display this data in a variety of ways.

In the universal evaluation selection screen, enter the following in the *Control* group box:

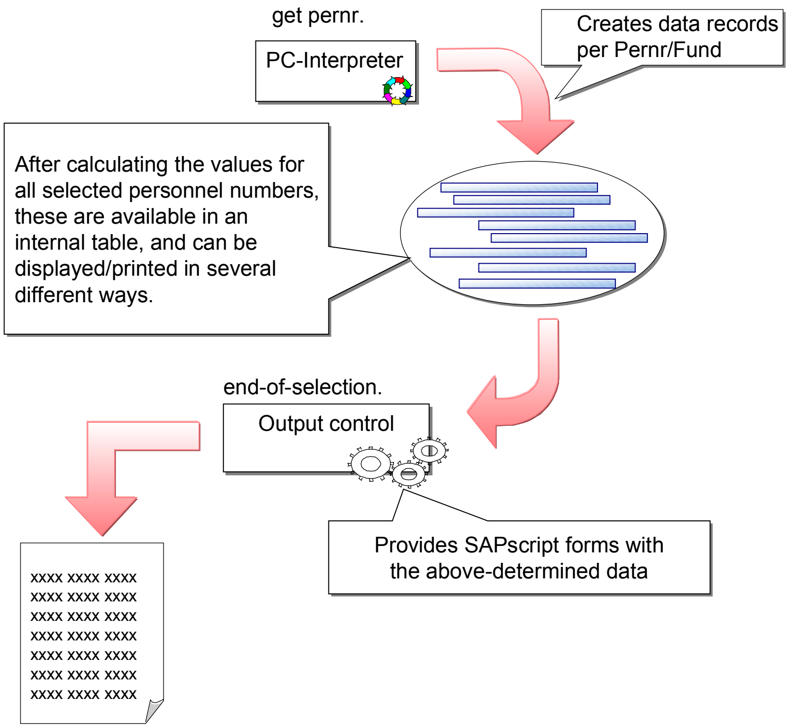
- In the *PC Schema* field, enter the processing type
- In the *Output Sequence* field, enter the output type, if required.

The PC interpreter creates and processes the data. The values determined for each personnel number and fund are transferred to a structure (see Output Assignments [Page 67]), and written to an internal table. You can use this data in several ways at the end-of-selection point in the report. That is, you can display the data on several different forms.

Errors during the processing schedule trigger the following procedures:

System error	Entire processing stops, a PC dump is displayed and, if necessary, sent per e-mail.
Application error (Returncode 1-3)	The employee is indicated as rejected in the notification list. Generated notifications can be displayed. Data transferred up until the shutdown is displayed. Batch-Input-records are written.
Application error (Returncode 4)	As above, except that the employee is removed from the display and no BI-records are written

The following diagram shows the basic operation of universal evaluations:


get pernr.
PC-Interpreter
Creates data records per Pernr/Fund
After calculating the values for all selected personnel numbers, these are available in an internal table, and can be displayed/printed in several different ways.
end-of-selection.
Output control
Provides SAPscript forms with the above-determined data
xxxx xxxx xxxx
xxxx xxxx xxxx
xxxx xxxx xxxx
xxxx xxxx xxxx
xxxx xxxx xxxx
xxxx xxxx xxxx
xxxx xxxx xxxx

If you have not activated the the option *direct SAPscript display*, the data is displayed in a list at the end of the processing.

The following functions are available in this list display, as in many reports in the SAP standard system:

- Summarizing
- Filtering
- Display variants (Arranging, Showing/Hiding fields)
- Exporting to Word or Excel
- Sending the list as an e-mail

PC Schemas

Definition

The PC schema determines — as a universal evaluation entry parameter — the type of processing.

In particular, it determines:

- The PC activity to be carried out by the interpreter,
- An output assignment type.

The output assignment type determines — together with the respective evaluation type — the output assignment.

Use

You can set the valid PC schemas for an evaluation type. In this way, you can control the possible entries and the consistency check on the selection screen.

The following settings are also possible for a PC schema:

- Eligibility of the option *Testmode* for PC-evaluation (general)
- Lock all accounts prior to the start of the processing (PF-specific)

> If you open the accounts with the operation *Attach to account list* (E3) with authorization level *W* or *X*, a database lock is set for the respective account, until the end of the individual processing. If a restart proves difficult, because the processing for some personnel numbers cannot be completed (for example, due to a locked account), you should lock all accounts. Note, however, that this prevents you from processing all accounts in this client.

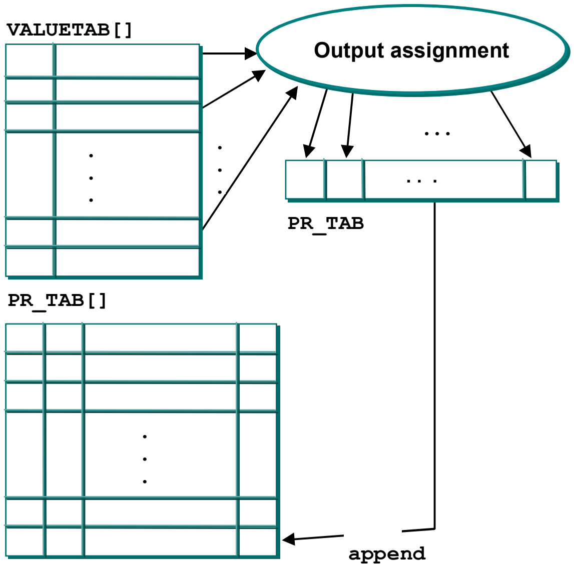
Output Assignments

Ausgabefelder der SAPscript-Auswertungen [Page 68]

Zusatzadresse [Page 71]

Definition

In order that the PC activity results (internal table `VALUETAB[]`) can be used by a universal evaluation, these values must be transferred to a structure (internal table header). Determine which value (identified by value type), is copied to which field in the output assignment.



The possible fields depend on the evaluation type. The SAP*script* evaluations are complex for creating subtotals. For technical reasons, this structure cannot be used for list evaluations. For this reason, the actual output assignment depends on the evaluation type, and — indirectly — on the PC schema.

SAP*script* Evaluation Output Fields

The field string for SAP*script* output has the following structure (Type P02VS_TAB_PRINTREC):

proc. type		
sumlv		
count		Number since the last control break
pers. no.		Personnel number
ahv no.		AHV number
ename		Name format (Last name, First name)
gender		Gender
Fund		Pension fund
comp. code		Company code
cont. area		Controlling area
pers. grp.		Person group
pers. subgrp.		Person subgroup
pers. area		Personnel area
pers. subarea		Personnel subarea
cost center		Cost center
fld01		Additional field for group formation
fld02		ditto
fld03		ditto
fld04		ditto
address	title	**Do not use!**
	title_pers	Person’s title
	name_pers	Name of the person: Title, first name, last name
	name1	**Do not use!**
	name2	**Do not use!**
	name co	c/o-Name
	lang.	Language
	str.	Street
	house num	House number
	pstlcd	Postal code
	pstlcd2	P.O. box postal code

	town01	Town
	town02	
	regio	Canton
	Ctry1	Country
	po box	P.O. box
	title	Title key
	titxt	Title text
data-	v0101	Packed number
	...	
	v0110	
	v0201	Packed number to 2 decimal places
	...	
	v0250	
	v0301	Packed number to 4 decimal places
	...	
	v0310	
data1-	v0401	12-character string
	...	
	v0410	
	v0501	Date
	...	
	v0515	
	v9901	Value of undefined type (as 12-CHAR)
	...	
	v9910	

These fields exist in the output table PR_TAB, and in the field string GRP_Wax, which contains the corresponding data for group x (x can be 1, 2, 3, 4, or 5, where 1 is the lowest control level, and 5 is reserved for the total).

Texts for the organizational units are also in the fields:

katxt	Text for the fund
pgtxt	Text for the person group (Active, Retiree,...)
pktxt	Text for the person subgroup

wktxt	Text for the personnel area (not language-dependent)
bttxt	Text for the personnel subarea (not language-dependent)
kstxt	Text for the cost center

Additional address

Apart from the employee’s usual address, you can specify an additional address in the universal-evaluation selection screen to be displayed on the modified SAP*script* forms. As not all PC forms support this possibility, you must confirm the additional address per output sequence. Set this up under *Evaluations -> Basic objects for SAPscript evaluations –> Define output sequence attributes*. You can read the data on a SAPscript form in the structure ADR2.

You can maintain addresses in the Central Address Maintenance (ZAV) in the group PA01. Access the address maintenance transaction from the universal evaluation selection screen, or with the ’Go to’ menu in the user interface.

SAP*script* Evaluations

Processing Control Forms

A *PC form* is a SAP*script* form, used in PC Universal Evaluations. Apart from the name of the corresponding SAP*script* form, you must define various PC-related characteristics for a PC form:

- In the SAPscript Editor initial screen (transaction SO10), you must specify the name of a SAPscript-Include for the header and footer (for example, HR_PF_HEADER) and for the address. Enter **ADRS** in the *Text-ID* field.

- You must define an indicator that separates list and single-sheet outputs. After displaying each data record (text element RECORD), insert a page break.

The addresses are included in the SAP*script* forms. These available fields can also be used in the address includes. You can format the addresses with the SAP*script*-command *ADDRESS*.

For forms that are to printed as letters, you can format the address for window envelopes. The PC forms in the standard system enable you to use the paragraph format *KL* (small) and *SK* (very small) for the printout.

Text Elements

SAP*script* Universal Evaluations correspond to the following SAP*script* form text elements (Window MAIN):

Textelement	Meaning
NP	Page break (with no list output)
RECORD	Data record; the current data is in the structure PR_TAB
GRSx	Group header x (x = 1,..., 4)
ZWSx	Subtotal for group x (x = 1,..., 5); the current data is in the structure GRP_Wax

Output Sequences

An *output sequence* indicates the SAP*script* output for each selection parameter or dialog box, and generally summarizes several PC forms – each with a group formation type – that, as a result of a one-time calculated amount of values, always follow sequentially. In this way, you can print out a sequence of insurance statements and cover letters at the same time.

Special case:
To print an individual PC form, you must define an integrated output sequence.

To control possible entries and the consistency check in the selection screen, you can define the compatibility of output sequences with evaluation types and PC schemas.

Group Formations

Use

You can define up to four groups for a SAP*script* list display by assigning a *group formation type* to an output step. The relevant sorting, display and formation of subtotals is done by the respective report.

Structure

You can specify up to four output structure fields for a group formation type that control the group formation, where the lowest level is 1, and the highest is 4. You can also use the pseudo-control level 5 to display a total.

You can display a header and a subtotal for each control level. You must indicate these in the SAP*script* form with the text elements GRSx and ZWSx.

The level x subtotal data is in the structure GRP_WAx.

⇨ You can also use group formation types to simply control sorting.

Creating Batch Input Sessions

For the universal evaluation to create a BI session, and to make available the corresponding parameters on the selection screen, you must indicate the respective PC schema in the setting *Evaluations –> Basic objects –> PC schemas –> Maintain PC schema settings* as being *BI-relevant*.

Universal evaluation exports the calculated values, whose value types are assigned to entry fields in the *Individual Data PF* (0279) infotype. The calculation reference date is used as the start date.

It is also possible to export additional infotype records per Batch-Input. This is controlled using tables T5CP4 and T5CP5.

If a required value type for the additional BI record is not defined for a personnel number, then this BI record is not exported.

Example: Salary increase for a benefit-based fund.

Here, you must recalculate the *Individual Values PF* (0279) infotype (new contributions), and specify an arrears payment (exceptional contribution) using the *Recurring Payments/Deductions* (0014) infotype.

To create infotype 0014 records with exceptional contributions for a change in salary, you must specify the sequential number to differentiate between the various infotype records in the view cluster for the corresponding PC schema.

For such a record, you must also determine how to fill the individual fields.

For the field names, call up the *Technical Info* function of the F1-Help for the relevant infotype field.

Additional Entry Parameters

In order to process data that is not stored in the system, in universal evaluations, you can define additional entry parameters for the universal evaluation selection screen, by assigning value types.

Example: Simulated calculations, especially entry simulations

Output Sorting for SAP*script* Universal Evaluation

A loop is run for each PC form of the respective output sequence for the entire PR_TAB[].

If you want to print several sheets per employee (for example, insurance statement with cover letter), you must do this with a SAP*script* form. You can specify the number of copies per personnel number in the setting *PC form attributes*. The text element *RECORD* is then displayed to correspond with this number in the *MAIN* window.

Within a SAP*script* form, you can select one of several provided output texts, by querying an indicator in the *MAIN* window.

In the standard system, the records created for output are sorted by personnel number. To implement a different sorting, the group formation type can be misused: Unavailable text elements for group headers or subtotals are ignored.

If you want to sort by gender and name, a subtotal formation takes place for the gender. You must then specify the field for the first control level ENAME, and specify GESCH for the second control level. Leave the text elements GRS1 and ZWS1 empty (no header or subtotal per name). Enter the relevant text for GRS2 and ZWS2.

Infotype Texts in SAP*script* Universal Evaluation

You can print the comments for the *Basic Data PF* (0278) infotype in a SAP*script* form as follows:

- The corresponding form receives a *NOTES* window. A paragraph format *NT* is defined, in which the comments lines are displayed.
- The *NOTES* window has a text element *HDG*, which is displayed as a header.

If these conditions are met, the SAP*script* universal evaluation displays the comments text in the *NOTES* window. Ensure that a paragraph is defined for each line.

You can define a standard text, such as *no comments*, for the text element *NO_DATA* in the *NOTES* window. The processing then takes place in exactly the same way, as if no valid text was found for the *Basic Data PF* (0278) infotype as of the reference date.

Include Entry Simulation in Menus

You can include applicant evaluation in menus with the transaction PACS.

The system searches for PF functions for an active fund, according to table T5CP2 (selected, if necessary, per dialog window), that use this report (RPLPKBC0).

The corresponding PF function is started for a clear result; otherwise, you must select from the dialog window.

PC Online Evaluation

Berechnungen [Page 84]

Sichten [Page 85]

Use

PC Online Evaluation is a tool that enables simple evaluations on the screen. As opposed to universal evaluations, you directly receive the result after selecting the corresponding calculation.

Features

You can call up PC Online Evaluation:

- From the pension fund infotypes
- From the user interface
- When creating manual postings

Online Evaluation is shown as a dialog window. The respective evaluation results are displayed in the lower table control.

A calculation is made, as soon as you call up the online evaluation, which is selected in context, using a value type – dependent on the schedule.

When you call up a new calculation, the corresponding values are also calculated and displayed.

In Customizing, you can define one or two parameters per calculation, which are entered in the PC Interpreter, upon execution. In this way, you can run simulations. You must take these parameters into account in the PC activity for the respective calculation. The PC operation *Pseudo posting* is especially helpful here.

Calculations

A calculation specifies the type of values to be determined. A PC activity and a view are assigned to a calculation – dependent on the PC schedule.

Views

Views allow you to select a subset of relevant values from a PC activity result. A quantity of value types is assigned to a view, depending on the schedule.

User Interface

Konfiguration [Page 87]

Use

The PF Administration user interface (transaction PACA) summarizes the various elements of the PF module in one program.

Features

The user interface enables:

- Direct input to the workflow inbox
- Call up of master data administration
- Call up of account maintenance
- A complex personnel selection
- Call up of PF functions (reports, variants, PF recalculations)
- Starting of various service programs (for example, payroll postings)
- (De)Activating event-linkage of the Toplevel-Workflows to process the *Basic Pay* (0008) infotype

Configuration

To configure the user interface, make the following settings in the PF IMG, in the section *User Interface*.

See also:

PF Functions [Page 88]

PF Recalculations [Page 89]

Event Linkage Control [Page 90]

Personnel Selections [Page 91]

Using External Programs [Page 92]

PF Functions

A significant task of the user interface is to offer a menu for the various PF-relevant functions that supports a simple data flow (for example, transfer of personnel numbers to reports).

Set up these functions as PF functions in Customizing, and integrate them into the user interface with function views. Carry out the integration in the PF-IMG under *User Interface –> PF Functions*.

A PF function can be a transaction (with transaction variant), a report (with report variant) or a PF recalculation.

PF Recalculations

If no workflows are included for the PF administration, PF recalculations in the user interface offers the option of changing master data during processes such as entry, change in pay. You must recreate an *Individual Values PF* (0279) infotype manually (for example, calling up the PF recalculation *Change in Pay*, using the date of the change in pay as the reference date). However, the new values are calculated according to processing schedule.

A PF recalculation consists of one if the assigned PC activities. The results of this recalculation are evaluated by the user interface, and – partly dependent on Customizing – are assigned to the data objects to be created.

An *Individual Values PF* (0279) infotype record must be created for a PF recalculation. In addition , you can create records for other infotypes using Customizing. If follow-on postings are created in the PC activity assigned to the PF recalculation, these records are also created. In this way, the created data is available for changes.

You can assign entry parameters to a PF recalculation. (However, this is not used in the SAP standard system.)

In a benefit based fund, the pension record is managed in infotype 0279. When posting an entry benefit, you must create a new infotype 0279 record. Do this with a PF recalculation, which receives the amount of the entry benefit as an additional parameter, and which prepares the new infotype 0279 and the entry benefit posting. However, it is recommended that you manage pension records in the account.

Event Linkage Control

In certain situations, you may not want to trigger a workflow. For example, if you want to modify all pay data at the start of the year per batch–input. In such cases, you can manually deactivate (and later reactivate) the workflow event linkage for processing changes in the *Basic Pay* (0008) infotype.

To do this, you must specify the type and ID of the corresponding workflow.

Personnel Selections

You can select personnel amounts from the user interface as follows:

- Call up the ad hoc query selection
- Run the *Complex Personnel Selection* report

If you use the query selection, be sure that the user is assigned to the correct functional area. In the SAP standard system the functional area /SAPQUERY/HR_CH_PA_01 is used for the global work area. This area is assigned to the user group /SAPQUERY/HS. You can assign other functional areas for each user parameter:

	User parameter	**Standard value**
Work area	AQW	G
User group	AQB	/SAPQUERY/HS
Functional area	AQS	/SAPQUERY/HR_CH_PA_01

You can also use the functional area /SAPQUERY/HR_CH_PA_01 for the ABAP- or ad hoc query itself.

The selection result is available in a personnel list of the user interface. These persons are transferred, if necessary, to PNP reports. You can started this from the user interface, or you can process them individually by selecting (double-clicking) the respective person.

Complex personnel selection is a PNP report that also has other PF-specific selection parameters. As with the query selection, the selection result is processed further.

Using External Programs

You can run PF recalculations and re-create accounts using the dialog modules HR_CH_PK_CALCI and HR_CH_PK_CRACC from external programs.

This is especially useful for dynamic actions (external performs). As an example, see Report RPUPKFC0.

Verarbeitungssteuerung: Detail

Die Grundidee der Verarbeitungssteuerung ist es, anhand eines Verarbeitungsplanes Werte zu berechnen oder zu verarbeiten. Hierbei stellen sich einige Fragen, die in den nachfolgenden Kapiteln beantwortet werden:

- Welche Einstellungen können für Wertarten [Page 94] vorgenommen werden?
- Welche Einstellungen gehören zu einem Verarbeitungsplan [Page 101]?
- Wie werden Tabellenwerte [Page 104] definiert bzw. gefunden?
- Wie arbeitet der VS-Interpreter [Page 111]?
- Wie kann erreicht werden, dass in einer konkreten Situation genau die erforderlichen Verarbeitungsschritte ausgeführt werden? (→ VS-Aktionen [Page 115], Untermethoden [Page 118])
- Wie kann die Online-Hilfe gesteuert werden? (→ WA-Doku [Page 232], VS-Wertemengen [Page 107])
- Welche elementaren Verarbeitungen stehen zur Verfügung? (→ VS-Operationen [Page 119])
- Wie werden gewisse zusammengehörige Gruppen von Operationen verwendet? (→ Verarbeitungsmodelle [Page 212])
- Wie werden Fehler [Page 227] behandelt?
- How can I locate and recognize errors? (→ PF Debugger [Page 230])
- Welche Objekte der Verarbeitungssteuerung können dokumentiert werden und wie? (→ Doku-Konzept [Page 232])
- Was ist zu tun, wenn die Standardoperationen nicht ausreichen? (→ User-Exits [Page 233])

Value Type Characteristics

Definition

It is a good idea to assign an additional attribute to value types. This attribute is called a *value type characteristic*.

Use

Value type characteristics enable you to use processing control operations for all value types with a specific characteristic. In this way, you can assign an *age-related credit* to all value types that describe incoming postings, and to form the total of all employee postings in the processing schedule, where the value type has this characteristic.

Structure

The assignment of characteristics to value types is schedule-dependent, and is therefore customer-specific. Assign them with the option *Value type characteristics* from the PC maintenance interface.

Special Value Type Characteristics

Various value type characteristics are not only used to control calculations in the processing schedule. Some have a specific meaning in certain programs. These include:

WT Characteristic	**Name**	**Meaning**
01	Post manually	You can only use value types with this characteristic for manual postings.
02	Account Closure	Used to automatically reset an account closure (not yet implemented).
03	Post as a negative	Paid out amounts must be posted as a negative. Assign corresponding value types (for example, advances) to WT characteristic 03; where a negative amount is defaulted for the manual posting.
04	Include in WT List (Dialog)	Only these value type entries are displayed in infotype 0279 in the list of additional values (for calculation in workflow or user interface).
05	Manual posting event	An event is only triggered for these value type (→ Triggering of Workflows), if a corresponding posting is *manually* created.

Meaning of Value Type Characteristics in the Standard System

The assignment of characteristics to value types influences many calculations, especially if certain totals are to be read from the account. If this assignment is incorrect, the subsequent calculation cannot be accurate.

The pre-defined value type characteristics each have a meaning that is partly determined through use in the processing schedules, and partly through the operations themselves. The most important value type characteristics are:

WT Characteristic	Name	Meaning/Use
11	EE contributions	All EE contributions (not Risk) • Minimum Leaving Benefit acc. to FLV Art. 17 • Posted EE contributions in the account in certain evaluations (Insured person's list, Statistics list)
12	ER contributions	All ER contributions (not Risk) • Posted ER contributions in the account in certain evaluations (Insured person's list, Statistics list)
14	Exceptional contribution	Total of all (EE+ER) arrears payments for covering funds (Benefit Plan) • Determining the outstanding exceptional contributions
15	Exceptional contribution EE	Total of EE arrears payments for covering funds (Benefit Plan) A pay raise results in an arrears payment. This must be posted as a negative, and calculated against the exceptional contributions to be paid later. The resulting total is the outstanding exceptional contributions that can, for example, be deducted from the leaving benefit • Outstanding exceptional EE contributions

16	Exceptional contribution ER	Total of ER arrears payments for covering funds (Benefit Plan) • Outstanding exceptional ER contributions
18	EE Entry benefits with interest	EE Entry benefits with interest (a type of pseudo-savings credit) • Interest calculation on entry benefits for minimum leaving benefit acc. to FLV Art. 17
21	Pension	All pensions • Total of all paid-out pensions for certain evaluations (Statistics List)
22	Advance	All Advances (EHO, Divorce) • Total of all paid advances
24	Pension credit	Pension increase (through premium or capacity utilization level change) in the benefit plan • Determines the current benefits (benefit plan)
25	Incoming Payment	All postings (Entry benefits, premiums, deposits), that result in an increase in benefits (EE+ER) but which are not contributions • Projection of total savings credit • Controls follow-on postings (for example, pension credits in the benefit plan) • Total of all incoming payments for certain evaluations (Insured person’s list, Statistics list)
26	EE incoming payment	All postings (Entry benefits, premiums, deposits), that result in an increase in benefits (EE) but which are not contributions; Advances are to be entered as negative EE incoming payments • Minimum Leaving Benefit acc. to FLV Art. 17 • Interest calculation on EE incoming payments (Account closure) • Total of EE incoming payments for certain evaluations (Insured person’s list, Statistics list)

Meaning of Value Type Characteristics in the Standard System

27	ER incoming payments	All postings (Entry benefits, premiums, deposits), that result in an increase in benefits (ER) but which are not contributions • Interest calculation on ER incoming payments (Account closure) • Total of ER incoming payments for certain evaluations (Insured person's list, Statistics list)
30	Savings credit	All postings (contributions, deposits, interest), that go into the total (EE+ER) savings credit • Savings credit • Leaving benefit (= Savings credit) after an account closure in the contribution plan
31	EE Savings credit	All postings (contributions, deposits, interest), that go into the EE savings credit • EE Savings credit
32	ER Savings credit	All postings (contributions, deposits, interest), that go into the ER savings credit • ER Savings credit
36	EHO Advance	EHO Advances • Interest calculation on EHO advances (if necessary, benefit plan)
37	EHO Advance with interest	EHO Advance with interest; Advances can be managed as a pseudo-savings credit (In the benefit plan, advances normally go into the savings credit as negative deposits and interest is then calculated, This calculation is therefore, only relevant for the benefit plan) • Pension reduction through EHO advance
38	Divorce Advance	Divorce Advances • Interest calculation on divorce advances (if necessary, benefit plan)

39	Divorce Advance with interest	Divorce advances with interest (at the same time as WT characteristic 37) • Pension reduction through divorce advance
41	Interest EE	All interest on EE postings • Total of EE interest for certain evaluations (Insured person’s list)
42	Interest ER	All interest on ER postings • Total of ER interest for certain evaluations (Insured person’s list)
43	Interest LOB	All interest on LOB postings • Total of LOB interest for certain evaluations (Insured person’s list)
44	Interest on EE Entry Benefits	Interest on entry benefits (for which interest has possibly already been calculated). (Read with WT characteristic 18) • Minimum Leaving Benefit acc. to FLV Art. 17
51	LOB Age-Related Credit	All postings that go into the LOB Age-Related Credit (LOB Entry Benefits, LOB Age-Related Credit, LOB interest) • LOB Age-Related Credit
52	LOB Incoming Payment	All incoming payments that go into the LOB Age-Related Credit • LOB incoming payments for certain evaluations (Insured person’s list) • Interest calculation on LOB incoming payments (Account closure) • Projection of LOB Age-Related Credit

Value Type Characteristics for Dialog

To control which value type characteristics are available to the user in the dialog (for example, as a filter in posting maintenance), use the dialog value type characteristic table, in which you can define which value type characteristics are available for each PC schedule (*PF-IMG –> Processing Control* → *PC Schedules* → *Value Type Characteristics for Dialog*). If this table is empty, you must display all value type characteristics assigned to the respective PC schedule.

Schedule Value Types

Use

You can assign additional settings to value types, depending on the schedule:

1. The indicator *Global* specifies that the value assigned to the value type is not to be deleted when you call up the PC Interpreter, as would normally be the case. For example, you can use this option to implement numerators for all processed personnel numbers.
2. The indicator *employee grouping from feature* specifies that the employee grouping must be determined for each feature for the current value type method, after which the methods in the processing schedule can be differentiated.
3. The setting *Feature to determine the employee grouping* or *Employee grouping* is important, depending on the *Employee grouping from feature* indicator.

Assign the settings with the option *Value type characteristics* from the PC maintenance interface.

"PF Switzerland " Value Types

A certain number of value types are already included in the new PF module. These value types result from the setup of the example schedules, or have been generally enhanced.

The SAP value types satisfy the following naming convention:

Prefix	Meaning
BG	Capacity utilization levels
BS	Contribution rates
BT	Contributions
BW	Present value factors
DA	Duration, Age
DT	Data
FL	Indicators
GW	Value limits
IG	Degree of Invalidity
KA	Paid-out capital amounts
KE	Paid-in/Calculated capital amounts
KN	Account types
KO	Coordination deductions
KR	Coordination deduction calculation rates
KS	Capital amount calculation rates
LE	General benefits
LN	Wages
RA	Paid-out pension
RE	Pensions
RS	Pension rates
UN	Undefined
VT	Value type
ZA	Numerator
ZI	Interest
ZS	Interest rates

➡ When creating customer value types, take note of the customer name space. Customer value types must be in the form ***#...*** or **# ..**, where **#** is a digit.

PC Parameter

Definition

The concept of the *PC-Parameter* (Processing Control Parameter) allows you to model **table values,** such as contribution rates, coordination deductions, and present value factors.

The main goal of PC parameter data-modeling was to minimize maintenance for this value. It is especially important to be able to access the same PC parameters from different schedules, such as LOB values. Therefore, you cannot model the PC parameters as schedule-dependent objects.

Use

A PC parameter complies with a *Class of corresponding value tables*, for example, *Savings contribution rate EE.* For the various funds, these value tables can be *separately structured*, and contain *varying values*.

PC parameters are each described by a value type. They are grouped in *parameter areas*. Within a parameter area, the configuration of a PC parameter (that is, assignment of key terms) and values is clear. In other words:

You can identify a PC parameter within a parameter area with a value table.

Structure

PC parameters are stored in a table with the following structure:

Value type	ParArea	Gender	Ref. Value 1	Ref. Value 2	End	Start	Value

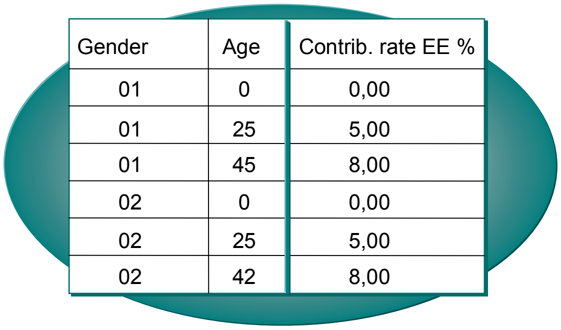
Here, you can configure the key terms Gender, Reference value 1 and Reference value 2:

- Reference value 1 with a *gender-specific* indicator
- Reference value 2 by assigning (or not) a value type.

In this way, you can control the table display in the maintenance interface.

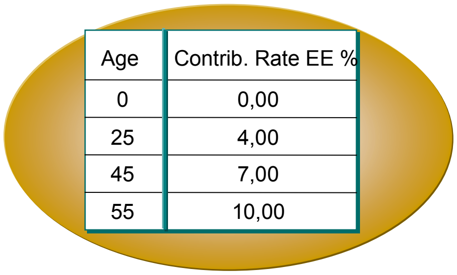
Also, a value table can exist for a PC parameter (indicated by a value type), within a parameter area, the form and contents of which are clear. For simplicity, the validity periods have been omitted from the following diagram:

Parameter area 1



Gender	Age	Contrib. rate EE %
01	0	0,00
01	25	5,00
01	45	8,00
02	0	0,00
02	25	5,00
02	42	8,00

Parameter area 2



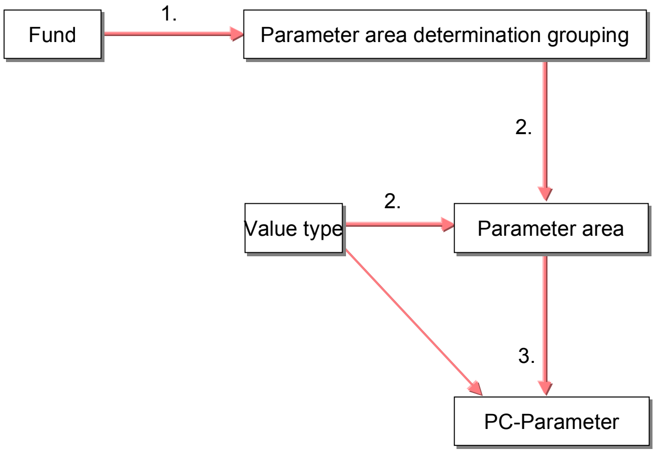
Age	Contrib. Rate EE %
0	0,00
25	4,00
45	7,00
55	10,00

How is the parameter area determined, which is used to determine the value type?

You must be able to assign a parameter area for each individual value type, and it should be specific to the fund (for the processing context *PF Switzerland*). If you use the schedule for this assignment, you must set up and maintain individual schedules in case several funds exist with the same rule, but have different contribution rates.

As the fund is not a processing control term, a new PC term is required that can you can assign to a fund. This is the parameter area determination grouping (*PaGru*).

The following diagram illustrates how a parameter value is determined. Fund and value type are given. Possible additional key terms are not shown:



1. The parameter area determination grouping (PaGru) is assigned to the fund. You must determine a parameter area for this PaGru and the given value type. The value type and

the parameter area determine the PC parameter. (that is, a value table), from which the desired value can be determined, using other key terms to be specified.

PC Value Sets

Definition

PC Value Sets enable you to form subsets with the data types from value type-specified character string sets. Use this to specifically control the F4-Help.

A value type that is to contain wage types, must be a *character string*, even though only a small number of all possible character strings can be used for a wage type.

Use

An ABAP subprogram must be available in the function group RPCH for each PC value set that:

- Checks whether a value is a set element, and
- Displays a selection dialog window of all set elements.

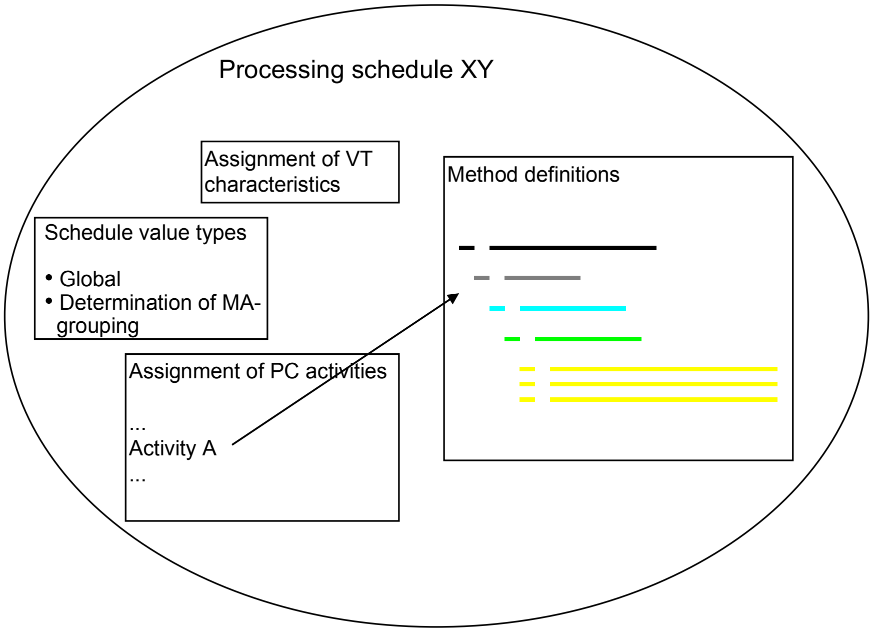
Customer enhancements are generally not required for PC value sets. If necessary, you can make enhancements by assigning parameters to existing value-set subroutines in the function group RPCH: the value sets are assigned a suffix to select the value set form mentioned above, and up to two further parameters, which are transferred when you call up the value set form. For example, you can process user-defined domains with fixed values or value tables (suffix of the value set form _DOM).

Schedule-Specific Objects

Apart from methods that form the actual schedule, there are other objects and settings that are specific to the respective schedule:

- The assignment of PC activities to schedule methods
- The assignment of value types and value type characteristics
- Certain value type settings

The following diagram shows all entities included under the schedule term:


Processing schedule XY
Assignment of VT characteristics
Method definitions
Schedule value types
• Global
• Determination of MA-grouping
Assignment of PC activities
...
Activity A
...

You access the above settings with the PC maintenance transaction (PAC0/PUCA).

In the maintenance screen *Process 'Swiss Pension Fund' PC Context*, you can:

- Get the PC schedule from client 000, using *Utilities*
- Store documentation for PC schedules, value types and operations
- Check consistency of the processing context (PC parameters, Activity assignment, Method variants)
- Call up the debugger for the respective PC context
- Maintain schedules and schedule-specific settings

Processing Steps and Methods

Definition

The processing schedule consists of processing steps that are organized by the key terms *Value type*, *Method type*, *EE grouping* and *Method variant*.

The result of processing steps that are the same as these key terms are called PC methods.

Use

A processing step consists of a line type and an operation (in some case – for example, for a submethod – this is a pseudo-operation). The latter has parameters and end conditions.

Significance of PC Editor Fields

You determine how the result of the operation is processed with the *line type* field.

You define which process takes place with the *Operation* field.

Select F1-Help in the *Operation* field for documentation on the respective chosen operation. If necessary, confirm this by pressing ENTER.

In the *Parameter* group box, the various operation parameters control the processing and the calculation of the operation. Use the radio buttons to determine whether the parameter is interpreted as a constant or a value type.

In the *Conditions* group box, you can control whether error/warning messages are created by setting an indicator in the *No message* field. Use the radio buttons to determine how the next line is to be processed, depending on the result of the operation or submethod.

Call Up the Interpreter

The PC Interpreter contains the following information when called up (Function module interface HR_CH_PK_VS_PERFORM):

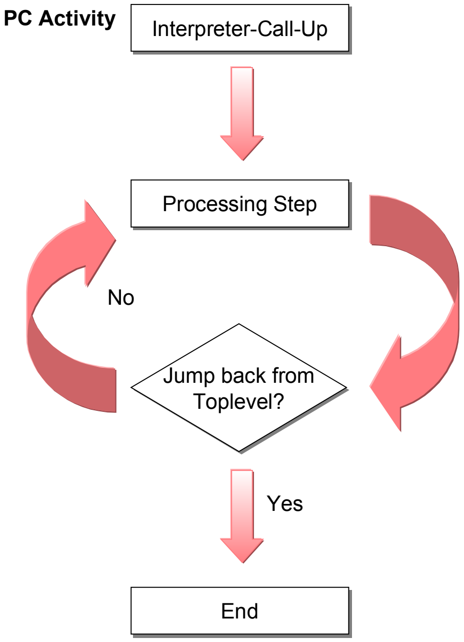
Parameter	Meaning
Reference date	Used by the interpreter to determine methods (for periodic delimitation); used to determine PC parameters; can be used in the processing schedule as a reference date for selections and calculations.
Start	Can be read in the processing schedule as the start of a period.
End	Can be read in the processing schedule as the end of a period.
Fund	Used to determine the assigned processing schedule for the reference date.
PC Activity	Specifies where the interpreter starts the processing in the PC schedule. That is, which PC method is used.
Personnel number	Needed to determine accounts.
Transaction class	Differentiates between employee (*A*) and applicant master data (*B*). If necessary, it is assigned to the field PME17-TCLAS (→ Features).
Indicator: *No error handling*	If this indicator is not selected, the PC interpreter does not display error handling, but simply shows a relevant return code.
Indicator: *No data base lock*	Controls processing lock for PF accounts, if you want to globally lock all accounts (universal evaluation option); first, set the lock externally
Indicator: *Test mode*	Prevents DB-Updates
Origin	Described the origin of postings.
Parameter table	Entry parameter, for example, for simulations
Value table	Table of values: If you want to keep values for several sequential interpreter call-ups, (for example, numerators), you must define the relevant value type (schedule-specific) as *global*. Entries with non-global value types are deleted from the table prior to processing.
Infotype tables	Already read master data – for example, by the logical database.

After processing, the determined values are in the value table. Also, information is transferred regarding errors that may have occurred.

The calling program determined which data is transferred as a parameter to the interpreter.

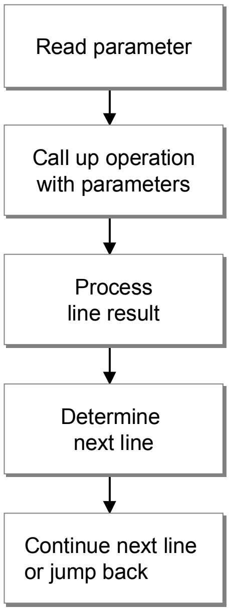
Interpreter Work Method

The following graphic shows the work method of the interpreter:



Processing Steps

Interpreter Work Method

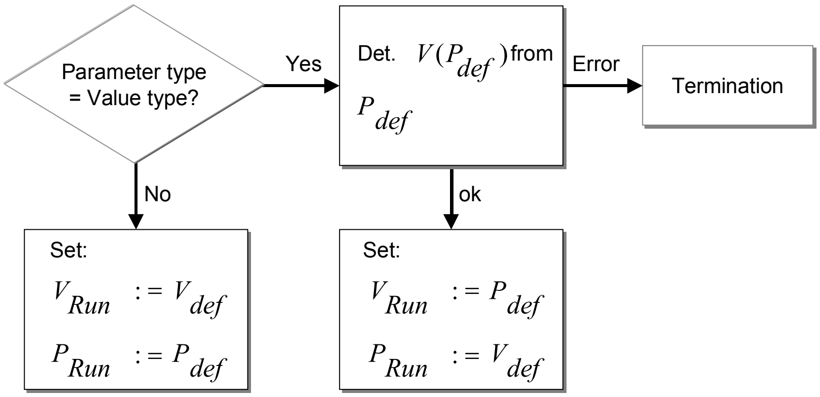

Read parameter
Call up operation with parameters
Process line result
Determine next line
Continue next line or jump back

The following shows these steps in detail.

The processing step parameters can be a *constant* or a *value type*. In the latter case, a value type is given as the parameter, where the current value of the operation is transferred.

Read parameter

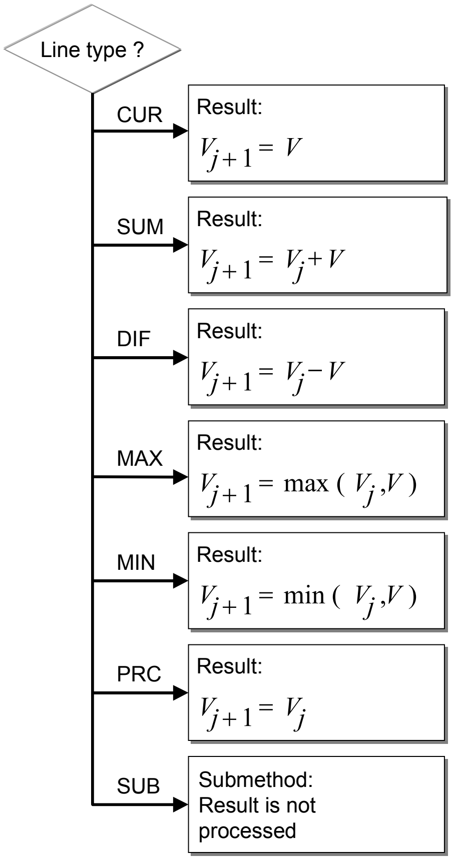
For each defined par. in the schedule (V_{def}, P_{def}) the run time parameter (V_{run}, P_{run}) is determiend.



The processing step operation is already complete. The result has been calculated and a return code given that corresponds to the to a condition regarding the operation run.

Process line result

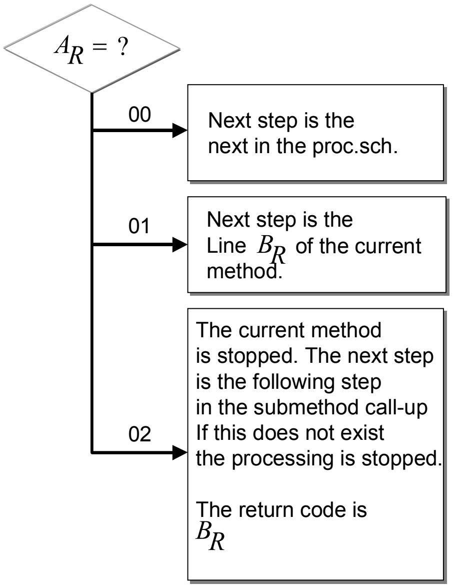
V is the value calculated by the called-up operation, V_j is the current value of the Curr. Calculated value type, V_{j+1} is the new value you want.



Interpreter Work Method

Determine next line:

R is the return code of the operation, A_R is the defined determination mode In the current processing step for the next processing step, B_R Is the corresponding parameter.


$A_R = ?$
00
Next step is the next in the proc.sch.
01
Next step is the Line B_R of the current method.
02
The current method is stopped. The next step is the following step in the submethod call-up If this does not exist the processing is stopped.
The return code is B_R

PC Editor

Definition

The PC Editor is a tool for maintaining method definitions of a processing schedule.

The methods of a schedule are displayed in a hierarchy:

- Value type (highlighted in white)
- Method type (highlighted in pink)
- EE Grouping (highlighted in light blue)
- Method variant (highlighted in green)
- Processing step (highlighted in yellow)

Use

Editor functions are generally used for objects (↔ nodes in the hierarchy display). To select the function place the cursor on the corresponding line.

The Editor has the following functions:

Display/Hide nodes	
Set focus	
Create	Create an element of the current hierarchy level. At least one processing line must exist for a value type (method type, EE grouping, method variant) in the processing schedule. This is initially created and can then be entered accordingly.
Copy	You can copy individual processing steps or sub-trees in the processing schedule.
Change	This changes the value assigned to a node (that is, value type, method type, EE grouping, method variant). You can edit the basic value set with the function *Maintain Values*.
Delete	You can delete individual processing steps or sub-trees in the processing schedule.
Change values	Change (menu) the value set for the currently selected hierarchy level in the processing schedule.
Create/Change marker	Set/change line marker in current line.
Delete marker	Delete line marker in current line. It can no longer be referenced.
Consistency check	The processing schedule is checked for consistency using various criteria.
Where-used list	You can determine the processing lines where value types and methods are used/called up.

Get method from schedule	You can load specified methods from a processing schedule to a specified value type (from the database). This is also possible for the current schedule, if methods have been accidentally deleted.
Complex search	Search processing lines according to such criteria as: • Method type • Value type • Operation • Parameter with value set This makes it possible to find all processing lines that use a particular operation.
Enter documentation	Enter implementation note for the value type methods.

A processing line must exist for each displayed object. If the Editor is started with an empty schedule, it inserts a default processing line, which can then be changed or — as soon as other methods exist — deleted. If necessary, subordinate objects are predefined with default values, when creating other new objects.

Method Assignment

Submethod skips within the processing schedule are entered in an “abstract” form: Skip from method type *nn* to value type *xxxx*.

The processing schedule methods are divided by more criteria; namely, employee grouping and method variant (validity period).

There are two variants per schedule value-type to determine the employee grouping that is controlled using schedule-specific settings for the value type:

1. Use a fixed employee grouping: This is the normal case, and is recommended for performance reasons.
2. Determine the employee grouping with a feature: In this case, the feature specified for the value type is evaluated when one of these wage type methods is called up. For an example, see the feature CHPK1.

To determine the method variant, the key date for each call-up of the PC interpreter is compared with the start and end dates.

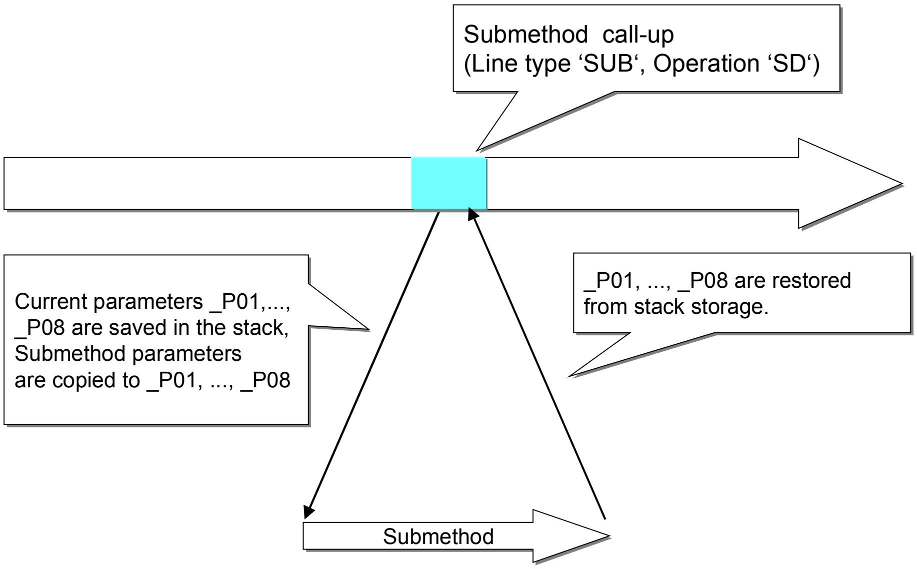
Submethods and Parameters

Definition

Submethods are an important instrument for the structuring and modularization of processing schedules.

Use

For a concrete processing, you must call up a row of methods in the correct sequence. That is, taking the dependencies into account, as a value can only be changed in one of the corresponding value type methods. There are special methods to do this, which only consist of a sequence of submethod jumps. (You can recognize these by the fact that they belong to value types with the form "_XYZ".) As the complete processes are also methods, you can use the various differentiation possibilities (method type, employee grouping, validity period).


Submethod call-up
(Line type 'SUB', Operation 'SD')
Current parameters _P01,..., _P08 are saved in the stack, Submethod parameters are copied to _P01, ..., _P08
_P01, ..., _P08 are restored from stack storage.
Submethod

This is reusable with submethod parameters. These are defined as a parameter of the submethod call up, and can be accessed in the called-up method with the pseudo-value types "_P01" to "_P08". Unlike the other value types, the parameters are not global. They are managed with stack storage („first in, first out"). The respective parameter type is dynamically determined.

The value type BSP0 contains the value *1234*. The submethod *Copy KEB0* with parameter 1 = *BSP0* (as value type) is called up. This submethod consists of one step. In this way, the value of BSP0 (*1234*) is assigned to the value type *KEB0*.

Overview of Operations

This section introduces the operations of the processing context for the Swiss pension fund, each with a short explanation.

Exceptions are an element of error handling within processing control.

Arithmetic

Addition (A1)	**Definition** Addition of two values by data type **Description** Result of the lines is the total of both parameters **Parameters** • First summand • Second summand **Prerequisites** The data types of the summands must be compatible. **Final conditions** 0 = o.k. **Exceptions** • Error (incompatible data types) • Addition of two values by data type	PKE_F_SUM
Subtraction (A2)	**Definition** Subtraction of two values by data type **Description** Result of the lines is the difference between both parameters **Parameters** • Subtrahend • Subtractor **Prerequisites** The data types of the parameters must be compatible. **Final conditions** 0 = o.k. **Exceptions** Error (incompatible data types)	PKE_F_DIFF

Multiplication (A3)	**Definition** Multiplication of two numerical values **Description** Result of the lines is the product of both parameters **Parameters** • First factor • Second factor **Prerequisites** The data types of the factors must be numerical. Final conditions 0 = o.k. **Exceptions** Error (incompatible data types)	PKE_F_PROD
Division (A4)	**Definition** Division of two numerical values **Description** Result of the lines is the quotient of both parameters **Parameters** • Dividend • Divisor **Prerequisites** The data types of the operands must be numerical. **Final Conditions** • 0 = o.k. • 1 – Division by 0 Exceptions Error (incompatible data types)	PKE_F_QUOT

Difference of two products (A5)	**Definition** Calculation of a term a * b – c * d **Description** Result of the lines is = p1* p2 - p3 + p4, where pj is the jth operation parameter. **Parameters** • Factor 1 • Factor 2 • Factor 3 • Factor 4 **Prerequisites** The data types of all parameters must be numerical. Final conditions 0 = o.k. **Exceptions** • Error in first product (incompatible data types) • Error in second product (incompatible data types) • Error in the difference	PKE_F_DIFF_PROD
Linear interpolation (A6)	Function value of a 1st grade polynom that is defined by two given markers, to one point **Description** Given markers are s1 = (x1,y1), s2 = (x2,y2), and a current value x. The result of the line is therefore: y = ((x2 - x) * y1 + (x - x1) * y2) / (x2 - x1) That is, y is calculated, so that (x,y) lies axis that runs through s1 and s2. **Parameter** x = current value x1 = x-coordinate of marker 1 y1 = y-coordinate of marker 1 x2 = x-coordinate of marker 2 y2 = y-coordinate of marker 2 **Prerequisites** The data types of all parameters must be numerical. **Final conditions** 0 – Value has been calculated 1 – Arithmetic error: that is, x1 = x2 **Exceptions** None.	PKE_LIN_IPOL

Round up or down to the nearest whole number (A7)	**Definition** Numerical value is rounded up or down to the nearest whole number with a rounding divisor **Description** The value of the current line is rounded to the nearest first parameter, according to the rounding divisor. The rounded value (no decimal point), is divisible by the rounding divisor. **Parameter** – value to be rounded – rounding divisor **Prerequisites** The value to be rounded must be numerical, the rounding divisor must be a whole number. Exceptions Error (data type)	PKE_ROUND
Rounding up (A8)	**Definition** Numerical value is rounded up with a specific rounding divisor **Description** The value of the current line is rounded up to the nearest first parameter, according to the rounding divisor. The rounded value (no decimal point), is divisible by the rounding divisor. **Parameters** • Value to be rounded • Rounding divisor **Prerequisites** The value to be rounded must be numerical, the rounding divisor must be a whole number. **Final conditions** 0 = o.k. **Exceptions** Error (data type)	PKE_ROUND_CL

Rounding down (A9)	**Definition** Numerical value is rounded down with a specific rounding divisor **Description** The value of the current line is rounded down to the nearest first parameter, according to the rounding divisor. The rounded value (no decimal point), is divisible by the rounding divisor. **Parameter** – value to be rounded – rounding divisor **Prerequisites** The value to be rounded must be numerical, the rounding divisor must be a whole number. **Final conditions** 0 = o.k. **Exceptions** Error (data type)	PKE_ROUND_FL
Complex total (AA)	**Definition** Total for all value types with specific attribute. **Description** Value of the current line is the total of all previously calculated values with a specific value type attribute. **Parameter** – Value attribute **Prerequisites** All values that are to form this total must already be calculated. **Final conditions** 0 = o.k. Exceptions Error (incompatible data types)	PKE_COMPLEX_SUM

Scale annual value (AB)	Definition Reduce by length of a given period. Description Value of the current line is the scaled annual value from the start and end of the specified period. Use the indicator 'Year with 360 days' to control whether you calculate exactly (365/366 days), or monthly (360 days). Result = P1 * (P3 – P2)/360 (or 364) Parameters – Annual value – Start of period – End of period – Year with 360 days Prerequisites The annual value relates to an entire year, for example, annual salary. Final conditions 0 = o.k. **Exceptions** None.	PKE_RECALC_TO_PERIOD

Arithmetic

Projected credit (AC)	**Definition** Value of age-related credit, at a foreseeable future point. Contributions and interest calculation are simulated from a given date to the end of the simulation. **Description** Value of the current line is the age-related credit resulting for an (estimated) specified retirement age. Included in the calculation are: 1. The previous annual credit (value at start of year) 2. The contributions to be paid according to the contribution rate (age-dependent) 3. Already-existing payments in the account as of the start of the year **Parameters** – Credit until the start of the year at the reference date – Current insured wage – Interest rate – Gender – Date of birth – End of simulation (generally the retirement date) – Value type for EE contribution rate – Value type for ER contribution rate – Value type attribute for incoming payments – Reference date: Determines the year when the simulation starts **Work method** The entry date is determined from infotype 0278. If this is not possible (for example, for applicants), the reference date is used for the entry date. The previous credit accrues interest for each year at the specified interest rate. In addition, the respective contributions are taken into account, as of the entry date, with the valid contribution rate for the respective age, and relative to the specified insured wage. In the retirement year, contributions and interest are reduced correspondingly. The values for the age-related credit are saved for the various ages in the iteration, and can subsequently be read using the operation *Projection value*. **Prerequisites** The transferred age-related credit is the valid value as of the start of the year of the reference date. **Final conditions** **0 = o.k.** **Exceptions** None.	PKE_GUTH_PROJ

Interest accumulation (AD)	**Definition**
	Value accrues interest at given rate until a specific date.
	Description
	Result of this operation is the amount of interest accumulated on specified capital at an interest rate in one period.
	Interest is accrued iteratively over the whole year from the start to the end of the period. The time between the start of the period and the 31.12. of the same year, and between the 1.1. of the year of the end of the period and the end of the period is calculated pro rata temporis.
	Parameters
	• Capital
	• Annual interest rate in %
	• Start of the period
	• End of the period
	Calculation
	`Interest payments = '1.00' + interest rate /100.`
	Incomplete start and end years are proportionately reduced:
	`Value = Value *('1.00'+ Interest rate/100*scale)`
	For all other years：
	`Value = Value * Interest payments`
	Final conditions
	0 = o.k.
	Exceptions
	None.
		PKE_INTEREST

Interest calculation for current posting (AE)	**Definition**: Value of the interest on the current posting from value date until end of the interest calculation. **Description** Result of this operation is the interest on the value of the current posting (processed with the operations *Get postings with value type*, *Get postings with VT attribute* or *Next Posting*) between the value date of this posting and the specified end date for the interest calculation. If the period from the value date of the posting and the end date of the calculation encompasses several years, the interest is compounded for each whole year. **Parameters** • Annual interest rate in % • Interest accrual until **Prerequisites** A current posting must exist. To do this, use one of the following operations: Get posting with value type, Get posting with VT attribute or Next posting. Final conditions 0 = o.k. **Exceptions** None.	PKE_INTEREST_ ENTRY

Earned benefit (BP) (AF)	**Definition** Earned benefit of an insured person's benefit (Base for leaving benefit in the benefit plan). **Description** The result of this operation is the earned benefit. It scales the maximum number of insured days from the entry to the leaving date, by the actual number of days in this period: Earned benefit = Insured benefit * (calculable insurance duration/possible insurance duration). **Parameters** • Insured benefit Normally: Retirement pension = Insured wage * Pension rate. • Entry date • Start of contribution payments (full insurance) • Leaving date • Retirement date **Prerequisites** The entry date must be prior to the leaving date. **Final Conditions** • 0 = o.k. • 1 = Value is not to be calculated **Exceptions** None.	PKE_ERW_LEIST

Projection value (AG)	**Definition** Age-related credit value at a later age; the projection must have been previously run “over and above this age”. **Description** The result is the value of the age-related credit at a (future) age that was previously calculated by the operation *Projected credit*. **Parameter** Age **Prerequisites** The operation *Projected age-related credit* must have been called up with a simulation end that at least corresponds to the specified age. **Final Conditions** • 0 = o.k. • 1 = No value found for the specified age **Exceptions** None.	PKE_GUTH_PROJ_VAL

Indicator: Year with 360 days (AH)

Definition
Controls the reduction of interest for periods of less than one year.

Description
If this indicator is set (default), then period-specific calculations, such as interest, are done on a monthly, rather than a daily basis.
The calculation procedure for daily differences is as follows:

- Daily
  This procedure uses the daily difference and, if necessary, is related to the number of days until the start date of the next year.
- Year with 360 days (banking procedure)
  The number of days is determined as 30 x monthly difference, in addition to the difference of the days. This includes the 31st of a month as well as 28th February. (Leap year: 29.02) is treated as the 30th.
  Example: Start End Daily difference
  01.01.1998 01.01.1998 1
  01.01.1998 02.01.1998 2
  30.01.1998 31.01.1998 1
  27.02.1998 28.02.1998 4
  27.02.1998 01.02.1998 5
  28.02.1998 01.03.1998 2
  27.02.2000 01.03.2000 5
  28.02.2000 01.03.2000 4
  (Leap year)
  28.02.2000 29.02.2000 3
  This value relates to 360 days.

Parameter
Value of the indicator (X or BLANK)

Exceptions
None.

Multiplication/Division (AI)	**Definition** Calculation of a * b / c **Description** Result of the lines is = p1* p2 / p3, where pj is the jth operation parameter. Use: If the values are large, there is the danger of overflow during separate processing (first multiplication, then division). In the opposite case, it is not exact enough. For internal calculations, this operation uses large enough fields, which prevents this problem occurring. **Parameters** • Factor 1 • Factor 2 • Divisor **Prerequisites** All parameters are to *two decimal places*. Final conditions 0 = o.k. **Exceptions** None.	PKE_MULT_DIV
Interest accumulation as of Date + 1 (AJ)	**Definition** Value accrues interest at given rate until a specific date. **Description** Result of this operation is the amount of interest accumulated on specified capital at an interest rate in one period. Interest is accrued iteratively over the whole year from the start to the end of the period. The time between the start of the period and the 31.12. of the same year, and between the 1.1. of the year of the end of the period and the end of the period is calculated pro rata temporis. This operation is similar to the interest accumulation operation, but starts accruing interest from the day after the value date. **Parameters** • Capital • Annual interest rate in % • Start of the period • End of the period **Final conditions** 0 = o.k.	PKE_INTEREST_D1

Interest for the current posting date + 1 (AK)	**Definition** Value accrues interest at given rate until a specific date. **Description** Result of this operation is the interest on the value of the current posting (processed with the operations *Get postings with value type*, *Get postings with value type attribute* or *Next Posting*) between the value date of this posting and the specified end date for the interest calculation. If the period from the value date of the posting and the end date of the calculation encompasses several years, the interest is compounded for each whole year. This operation is similar to the interest calculation for current posting operation, but starts accruing interest from the day after the value date. **Parameters** • Annual interest rate in % • Interest accrual until **Prerequisites** A current posting must exist. To do this, use one of the following operations: *Get posting with value type, Get posting with VT attribute* or *Next posting*. **Final conditions** 0 = o.k. Exceptions None.	PKE_INTERES_ENTRY_D1

LOB

LOB pay in period (B0)	**Definition** Determines the LOB pay in a period from a LOB base, taken from the PF account, infotype 0279, or payroll results. **Description** LOB-coordinated periodic wage relating to the annual value LOB wage is determined from the LOB base. This is read • From the account (the total of postings with value type LNB0 and posting date in the specified period), or, • From infotype 279 (the normal method), or • From the payroll results. If the LOB base is determined from the account, periods are created using infotype 278, which scales and coordinates the total value for these periods. Otherwise, periods are created in line with infotype 279. If possible, the LOB base is determined within a period from the infotype. Otherwise, the LOB base is read from the corresponding payroll results in the appropriate periods. If the maximum LOB wage is not reached in the individual periods, a "distribution wage type" amount can be spread over these periods (for example, 13th monthly wage). **Parameters** • Start of the period • End of the period The values read are always adjusted to the period length. If the period is 2 years, for example, the result is the LOB wage for 2 years. • Definitive wage value type This value type determines the insured person's LOB liability. If an employee is not liable, the value of this value type is zero in infotype 279. • LOB base wage type If the LOB wage is to be determined during the payroll run, the LOB base is read from this wage type. LOB distribution wage type	BVE_ANSAL

LOB coordination deduction (B1)	**Definition** Value of the LOB coordination deduction. **Description** Result of the operation is the LOB coordination deduction on the specified reference date. **Parameter** – Reference date **Prerequisites** The LOB coordination deduction KOB0 must be maintained as a PC parameter in parameter area BV for the specified reference date. **Final conditions** 0 = o.k. **Exceptions** Error accessing PC parameter	BVE_GET_ANKOORD
LOB minimum (B2)	**Definition** Value of the minimum insured wage according to LOB **Description** Result of the operation is the valid minimum LOB wage on the specified reference date. **Parameter** – Reference date **Prerequisites** The minimum LOB wage GWB0 must be maintained as a PC parameter for the specified reference date. **Final conditions** 0 = o.k. Exceptions Error accessing PC parameter	BVE_GET_ANMIN

LOB maximum (B3)	**Definition** Value of the maximum insured wage according to LOB **Description** Result of the operation is the valid maximum LOB wage on the specified reference date. **Parameter** Reference date **Prerequisites** The maximum LOB wage GWB1 must be maintained as a PC parameter for the specified reference date. **Final conditions** 0 = o.k. **Exceptions** Error accessing PC parameter.	BVE_GET_ANMAX

Coordinate according to LOB **(B4)**	Definition Coordination of a wage according to LOB **Description** Result of the operation is the wage coordinated in line with LOB. Minimum/Maximum LOB wage and LOB coordination deductions are adjusted in line with the period length. Parameters • Start of period Default: Current date • End of period Default: Period end is selected so that the period length is 1 year. • LOB degree of invalidity Degree of invalidity according to LOB (only 50% and 100% according to LOB, Section 24) • LOB Base Value to be coordinated **Calculation** Result = (LOB Base/Degree of Invalidity – coordinated wage*scaling) * Degree of Invalidity LOB minimum < Result < Forced LOB maximum **Final conditions** 0 = o.k. **Exceptions** Error accessing PC parameter.	BVE_KOORD

Degree of Invalidity according to LOB (B5)	**Definition** Degree of invalidity according to LOB **Description** Result of this operation is the degree of invalidity in line with LOB. The real degree of invalidity is used. As of 1997, and according to LOB, Section 24, the LOB degree of invalidity is defined as: • 50%, if the employee has at least a 50% disability • 100%, if the employee has at least a 2/3 disability **Parameters** – Reference date – Real degree of invalidity **Prerequisites** The LOB degree of invalidity IGB0 must be maintained as a PC parameter in parameter area BV. **Final conditions** 0 = o.k. Exceptions Error accessing PC parameter.	BVE_INVGR

LOB Age-Related Credit (B6)	**Definition** LOB age-related credit **Description** Result of the operation is the LOB coordination deduction on the specified reference date. **Parameter** Reference date **Prerequisites** The LOB coordination deduction KOB0 must be maintained as a PC parameter in parameter area BV for the specified reference date. **Final conditions** 0 = o.k. **Exceptions** None.	BVE_AGUTH
Retirement date (B7)	**Definition** Retirement date **Description** The operation determines the retirement date from the view *Retirement Age Calculation* (V_T5CR0, subapplication *PKBV*). **Parameters** – Date of birth – Gender **Return value** – Retirement date **Final conditions** 0 = o.k. 1 = no valid entry found in T5CR0 2 = invalid entry in T5CR0 3 = error during special processing 4 = internal error **Exceptions** None.	BVE_RETIREMENT_DATE

Minimum leaving benefit (B9)	**Definition** Minimum Leaving Benefit acc. to FLV, Section 17 **Description** Result of this line is the minimum value of the leaving benefit in line with Section 17, FLV. It is defined as "earned entry benefits with interest and ... the (insured person's) paid contributions including a bonus of 4% as of age 20, but not exceeding 100%". Until the start of the simulation, entry benefits (VT attribute 26) are read, interest is added (VT attribute 44), and employee contributions (VT attribute 11). Interest is calculated for the total of all entry benefits with interest between the start of simulation and leaving using the interest rate that is calculated from the specified value type. Contributions are added for each month in this period. The simulation is not exact: It assumes a constant contribution, which must be specified as an operation parameter. Possible future changes to master data are not taken into account. It is also assumed that possible entry benefits accrue interest in the account up until the start of the simulation. In the standard system, the minimum leaving benefit is calculated in value type KAB5. **Parameters** • Date of birth • Gender Currently not required. • Value type for interest rate During the simulation, the interest rate calculated here is used to calculate interest for the total entry benefits from the start of the simulation to the leaving date. • Monthly EE contribution Basis for the simulation • Leaving date • Simulation from After this date, the basic calculation data is not read from the account, but is projected per simulation. **Exceptions**	BVE_MIN_AUL

Retirement age LOB (BA)	**Definition** LOB retirement age **Description** The operation calculates the age at the time the pension starts using the table *Retirement Age Calculation* (T5CR0, subapplication *PKBV*). **Parameters** – Reference date – Age – Reference date **Return value** – Retirement age **Final conditions** 0 = o.k. 1 = no entry found in T5CR0 2 = invalid entry in T5CR0 3 = error during special processing 4 = internal error Exceptions Error accessing PC parameter.	BVE_RETIREMENT_DATE

Proj. age-related credit with interest (BB)	**Definition** Projected LOB age-related credit (with interest → Base for calculating retirement pension) **Description** Value of the current line is the LOB age-related credit resulting from the interest accrual for the retirement date in line with LOB. **Parameters** • LOB age-related credit until reference date • Insured wage • Gender • Date of birth • Entry date • Reference date **Working method** The interest from the value date to the year-end (reference date) is determined for postings with the specified VT attribute. This value is used to correct subsequent annual interest calculations. The previous credit accrues interest for each year at the specified interest rate. In addition, the respective contributions are taken into account, (as of the entry date, at the earliest), with the valid contribution rate for the respective age, and relative to the specified insured wage. In the retirement year, contributions and interest are reduced correspondingly. **Prerequisites** The transferred age-related credit is the valid value as of the reference date. **Exceptions** • Error determining retirement age (PC parameter) • Error determining interest rate (PC parameter)	BVE_AGUTH_PMZ

Proj. age-related credit without interest (BC)	**Definition** Projected LOB age-related credit (without interest → Base for calculating invalidity pension) **Description** Value of the current line is the LOB age-related credit without interest accrual for the retirement date in line with LOB. **Parameters** • LOB age-related credit until reference date • Insured wage • Gender • Date of birth • Entry date • Reference date **Working method** In addition, the respective contributions are taken into account, (as of the entry date, at the earliest), with the valid contribution rate for the respective age, and relative to the specified insured wage. In the retirement year, contributions and interest are reduced correspondingly. Prerequisites The transferred age-related credit is the valid value as of the reference date. **Exceptions** Error determining retirement age (PC parameter)	BVE_AGUTH_POZ

Retirement pension (BD)	**Definition** LOB retirement pension **Description** Result of the operation is the LOB retirement pension, resulting from the specified LOB age-related credit with projected interest. **Parameter** Projected LOB age-related credit with interest **Prerequisites** The LOB retirement pension rate must be maintained as a PC parameter in parameter area BV. **Calculation** Result = Projected age-related credit with interest * RSB0/100 **Final conditions** 0 = o.k. Exceptions Error accessing PC parameter	BVE_ARENTE

Invalidity pension (BE)	**Definition** LOB Invalidity pension **Description** Result of the operation is the LOB invalidity pension, resulting from the specified LOB age-related credit without projected interest. **Parameter** Projected LOB age-related credit without interest **Prerequisites** The LOB invalidity retirement pension rate must be maintained as PC parameter RSB1 in parameter area BV. Calculation Result = Projected age-related credit without interest * RSB1/100 **Final conditions** 0 = o.k. **Exception** Error accessing PC parameter	BVE_IVRENTE
Orphans pension (BF)	**Definition** LOB Orphans pension **Description** Result of the operation is the LOB orphan pension, resulting from the specified LOB retirement pension. **Parameter** LOB retirement pension **Prerequisites** The LOB orphan pension rate must exist as PC parameter RSB2 in parameter area BV. **Calculation** Result = LOB retirement pension * RSB2/100 **Final conditions** 0 = o.k. Exceptions Error accessing PC parameter	BVE_WARENTE

Widows pension (BG)	**Definition** LOB Widows pension **Description** Result of the operation is the LOB widow's pension, resulting from the specified LOB retirement pension. **Parameters** • LOB retirement pension • Gender This pension is only paid out to women. **Prerequisites** The LOB widow's pension rate must exist as PC parameter RSB3 in parameter area BV. **Calculation** Result = LOB retirement pension * RSB3/100 **Final conditions** 0 = o.k. Exceptions Error accessing PC parameter	BVE_WIRENTE

Invalidity-Child pension (BH)	**Definition** LOB Invalidity-Child pension **Description** Result of the operation is the LOB invalidity-child pension, resulting from the specified LOB retirement pension. **Parameters** • LOB retirement pension **Prerequisites** The LOB invalidity-child pension rate must exist as PC parameter RSB4 in parameter area BV. **Calculation** Result = LOB retirement pension * RSB4/100 **Final conditions** 0 = o.k. Exceptions Error accessing PC parameter	BVE_IKRENTE
Retired-Child pension (BI)	**Definition** LOB Retired-Child pension **Description** Result of the operation is the LOB retired-child pension, resulting from the specified LOB retirement pension. **Parameters** • LOB retirement pension **Prerequisites** The LOB retired-child pension rate must exist as PC parameter RSB5 in parameter area BV. **Calculation** Result = LOB retirement pension * RSB5/100 **Final conditions** 0 = o.k. Exceptions Error accessing PC parameter	BVE_PKRENTE

Share of entry benefit to minimum leaving benefit (BJ)	**Definition** Total of entry benefits as part of the minimum leaving benefit according to FLV, Section 17 **Description** Result of this line is the total leaving benefits within the leaving benefit in line with Section 17, FLV. The calculation corresponds to that in the operation *Minimum Leaving Benefit LOB*. **Parameters** • Date of birth • Gender Currently not required. • Value type for interest rate During the simulation, the total entry benefits accrue compound interest with the determined interest rate from the start of the simulation to the leaving date. • Monthly EE contribution Basis for the simulation • Leaving date • Simulation from After this date, the basic calculation data is not read from the account, but is projected per simulation. **Calculation** The calculation corresponds to that in the operation B9 with entry benefits = KEB5. **Final Conditions** • 0 = o.k. • 1 -> cannot be determined **Exceptions** Error determining interest rate (PC parameter)	BVE_MAUL_EINK

Share of LMB interest to minimum leaving benefit (BK)	**Definition** Total of interest on entry benefits as part of the minimum leaving benefit according to FLV, Section 17. **Parameters** • Date of birth • Gender Currently not required • Value type for interest rate During the simulation, the total entry benefits accrue compound interest with the determined interest rate from the start of the simulation to the leaving date. • Monthly EE contribution • Basis for the simulation • Leaving date • Simulation from After this date, the basic calculation data is not read from the account, but is projected per simulation. **Calculation** The calculation corresponds to that in the operation *B9* with LMB interest = ZIB2. **Exceptions** Error determining interest rate (PC parameter)	BVE_MAUL_ZINS

Share of EE contributions to minimum leaving benefit (BL)	**Definition** Total of EE contributions as part of the minimum leaving benefits according to FLV, Section 17. **Description** Result of this line is the total interest on entry benefits within the leaving benefit in line with Section 17, FLV. The calculation corresponds to that in the operation *Minimum Leaving Benefit LOB*. **Parameters** • Date of birth • Gender Currently not required • Value type for interest rate During the simulation, the total entry benefits accrue compound interest with the determined interest rate from the start of the simulation to the leaving date. • Monthly EE contribution Basis for the simulation • Leaving date • Simulation from After this date, the basic calculation data is not read from the account, but is projected per simulation. **Calculation** The calculation corresponds to that in the operation *B9* with EE contribution = BTB5. **Final Conditions** • 0 = o.k. • 1 - cannot be determined **Exceptions** Error determining interest rate (PC parameter)	BVE_MAUL_ANBT

Number of years for age-related bonus to minimum leaving benefit (BM)	**Definition** Number of years to be calculated (min {25,max {Age – 20,0}}) for the age-related bonus of the minimum leaving benefit acc. to FLV Art. 17. **Description** Result of the line is the number of years upon which the calculation of the age-related bonus in the leaving benefit is based, in line with LMB, Section 17. The calculation corresponds to that in the operation *Minimum Leaving Benefit LOB*. **Parameters** • Date of birth • Gender Currently not required • Value type for interest rate During the simulation, the total entry benefits accrue compound interest with the determined interest rate from the start of the simulation to the leaving date. • Monthly EE contribution Basis for the simulation • Leaving date • Simulation from After this date, the basic calculation data is not read from the account, but is projected per simulation. **Calculation** The calculation corresponds to that in the operation *B9* with number of years = DAB3. **Final Conditions** 0 = o.k. 1 = cannot be calculated **Exceptions** Error determining interest rate (PC parameter)	BVE_MAUL_JAHRE

Age-related bonus rate to minimum leaving benefit (BN)	Percentage (= number of years * 4%) for the age-related bonus of the minimum leaving benefit acc. to FLV, Section 17 **Description** Result of this line is the rate for calculating the age-related bonus within the leaving benefit in line with Section 17, FLV. The calculation corresponds to that in the operation *Minimum Leaving Benefit LOB*. **Parameters** • Date of birth • Gender Currently not required • Value type for interest rate During the simulation, the total entry benefits accrue compound interest with the determined interest rate from the start of the simulation to the leaving date. • Monthly EE contribution Basis for the simulation • Leaving date • Simulation from After this date, the basic calculation data is not read from the account, but is projected per simulation. **Calculation** The calculation corresponds to that in the operation *B9* with age-related bonus rate = KSB6. **Final Conditions** 0 = o.k. 1 = cannot be calculated **Exceptions** Error determining interest rate (PC parameter)	BVE_MAUL_AZAULS

Share of age-related bonus to minimum leaving benefit (BO)	**Definition** Age-related bonus (= percentage * total of EE contributions) as part of the minimum leaving benefit acc. to FLV, Section 17. **Description** Result of this line is the age-related bonus in the leaving benefit in line with Section 17, FLV. The calculation corresponds to that in the operation *Minimum Leaving Benefit LOB*. **Parameters** • Date of birth • Gender Currently not required • Value type for interest rate During the simulation, the total entry benefits accrue compound interest with the determined interest rate from the start of the simulation to the leaving date. • Monthly EE contribution Basis for the simulation • Leaving date • Simulation from After this date, the basic calculation data is not read from the account, but is projected per simulation. **Calculation** The calculation corresponds to that in the operation *B9* with share of age-related bonus = KAB6. **Final conditions** 0 = o.k. **Exceptions** Error determining interest rate (PC parameter)	BVE_MAUL_AZAUL

Share of interest on EE contributions for the calculation of minimum leaving benefits acc. to section 17 of the FLV	**Definition** Minimum Leaving Benefit acc. to FLV, Section 17 **Description** The result of the line is the interest on employee contributions acc. to FLV, section 17. The calculation corresponds with the calculation in operation B9. In the SAP standard system, the operation writes the result to the value type ZIB3. **Parameters** • Date of birth • Gender (currently unnecessary) • Value type for interest rate Using the interest rate calculated here, during the simulation interest is calculated for the total of the entry benefits between the start of simulation and leaving. • Monthly EE contribution Basis for the simulation • Leaving date • Simulation from After this date, the basic calculation data is not read from the account, but is projected per simulation. **Prerequisites** None. **Final conditions** 0 = o.k. 1 = cannot be determined • Run time When calling one of the following operations for the first time, the run time is medium to high: Minimum leaving benefit • Total of entry benefits to minimum leaving benefit LOB • Interest on entry benefits to minimum leaving benefit LOB • Total of ER contributions to minimum leaving benefit LOB • Interest on ER contributions for minimum leaving benefit LOB	BVE_MAUL_ZINS_BT

Calendar Dates

Monthly difference (D0)	**Definition** Number of months between two dates **Description** Result of the operation is the number of whole months within the specified dates. **Parameters** • Date 1 • Date 2 **Examples** There are no months between 01.04.1995 and 30.04.1995. There is one month between 30.04.1995 and 01.05.1995. **Final Conditions** • 0 – Difference is not negative (Date 1 >= Date 2) • 1 – Difference is negative (Date 1 < Date 2) **Exceptions** None.	PKE_MN_DIFF
Annual difference (D1)	**Definition** Number of whole years between two dates **Description** Result of this operation is the difference between two specified annual calendar dates (Date 1 - Date 2). **Parameters** • Date 1 • Date 2 **Final Conditions** • 0 – Result is not negative (Year from date 1 >= Year from date 2) • 1 – Result is negative (Year from date 1 < Year from date 2) **Exceptions** None.	PKE_YR_DIFF

Calendar Dates

Start of next month (D2)	**Definition** Date at the start of the next month **Description** Result of the operation is the date of the start of the next month, after the specified date. **Parameter** Date Final conditions 0 = o.k. **Exceptions** None.	PKE_NEXT_MONTH
End of the month (D3)	**Definition** Date at the end of the current month **Description** Result of the operation is the date of the end of the month in which the specified date lies. **Parameter** Date Final conditions 0 = o.k. **Exceptions** None.	PKE_END_OF_MONTH
Start of the month (D4)	**Definition** Date at the start of the current month **Description** Result of the operation is the date of the start of the month in which the specified date lies. **Parameter** Date Final conditions 0 = o.k. **Exceptions** None.	PKE_BEG_OF_MONTH

Start of next year (D5)	**Definition** Date at the start of the next year **Description** Result of the operation is the 01.01, after the specified date. **Parameter** Date **Final conditions** 0 = o.k. **Exceptions** None.	PKE_NEXT_YEAR
End of the year (D6)	**Definition** Date at the end of the current year **Description** The year-end date after the specified date. **Parameter** Date **Final conditions** 0 = o.k. **Exceptions** None.	PKE_END_OF_YEAR
Start of the year (D7)	**Definition** Date at the start of the current year **Description** Result of the operation is the 01.01, in the year of the specified date. **Parameter** Date **Final conditions** 0 = o.k. **Exceptions** None.	PKE_BEG_OF_YEAR

Combine date (D8)	**Definition** Assign explicit values to a date component **Description** The result of this operation is a date, generated from the specified date, where day, month and/or year are replaced. If one of the parameters (day, month, year) zero, the original date does not change. **Parameters** • Date • Day • Month • Year **Final conditions** 0 = o.k. **Exceptions** None.	PKE_COMPOSE_DATE
Month from date (D9)	**Definition** Month in a date **Description** Result of the operation is the month of the specified date. **Parameter** Date **Final conditions** 0 = o.k. **Exceptions** None.	PKE_MONTH_OF
Year from date (DA)	**Definition** Year in a date **Description** Result of the operation is the year of the specified date. **Parameter** Date **Final conditions** 0 = o.k. **Exceptions** None.	PKE_YEAR_OF

Display date (DB)	**Definition** Date in display format (character string) **Description** Result of the operation is the formatted character string of the specified date. **Parameter** Date **Final conditions** 0 = o.k. **Exception** Error converting to display format	PKE_PRINT_DATE
Test: Is the date within the period? (DC)	**Definition** Test: Is d1 <= d <= d2 ? **Description** The operation checks whether the date lies between certain upper and lower limits, and sets the appropriate final condition. **Parameters** • Date • Lower limit • Upper limit **Final Conditions** • 0 – Date is within the period • 1 – Date is not within the period **Exceptions** None.	PKE_DATE_BETWEEN

Calendar Dates

Leaving date PF (DD)	**Definition** Fund leaving date (validity end date of the last infotype 0278 record) **Description** The result of this operation is the leaving date of the current employee, if he or she has left. It results from the *Leaving Date* field of an infotype 0278 record for the corresponding fund, with the insurance agreement that starts no later than the current reference date. If no infotype record is found, the last available record for the current fund is selected. Prerequisites • An infotype 0278 record for the current fund must exist for the personnel number. • Infotype 0278 must be delimited to the leaving date when the person leaves the pension fund. **Final Conditions** • 0 – Insured person has not yet left • 1 – Insured person has left **Exception** No infotype 0278.	PKE_LEAVE_DATE
Previous day's date (DE)	**Definition** Date – 1. **Description** Result of the operation is the day prior to the specified date. **Parameter** Date **Final conditions** 0 = o.k. **Exceptions** None.	PKE_DAY_BEFORE

Next day’s date (DF)	**Definition** Date +1. **Description** Result of the operation is the next day’s date, after the specified date. **Parameter** Date **Final conditions** 0 = o.k. **Exceptions** None.	PKE_DAY_AFTER
Start of full insurance cover (DG)	**Definition** Start of full insurance cover. **Description** The result of this operation is the start date for full insurance coverage, due to the insured person’s age. It is determined as follows: 1. Year = Year of birth + Age for the start of full insurance 2. Month = Parameter month (usually = 01) 3. Day = 01 **Parameters** • Date of birth • Age at start of full insurance • Month at start of full insurance **Final conditions** O.k. **Exceptions** None.	PKE_BEG_VOLLVER

Postings

Get postings (VT) (E0) (**E0**)	**Definition** Select postings by value date and value type **Description** This operation selects postings from the account, previously imported using the operation *Generate Account List*, by value date and value type. The operation value is the value of the first selected posting. You can process the selected postings in a loop with *Next posting*. **Parameters** • Value date from • Value date to • Value type • All personnel numbers? If this indicator is set, all postings read from the database are considered for selection, regardless of personnel number. Usually, only the postings for the current personnel number are processed. **Prerequisites** An account list must have been generated. **Final Conditions** • 0 – Postings were selected • 1 – No corresponding postings exist **Exceptions** Postings not yet loaded	PKE_READ_ACCNT

Get postings with posting date (VT) (E5)	**Definition** Select postings by posting date and value type **Description** This operation selects postings from the account, previously imported using the operation *Generate Account List*, by value date and value type attributes. The operation value is the value of the first selected posting. You can process the selected postings in a loop with *Next posting*. **Parameters** • Value date from • Value date to • Wage type attribute • All personnel numbers? If this indicator is set, all postings read from the database are considered for selection, regardless of personnel number. Usually, only the postings for the current personnel number are processed. **Prerequisites** An account list must have been generated. **Final Conditions** • 0 – Postings were read • 1 – No corresponding postings exist **Exceptions** Postings not yet loaded	PKE_READ_ACCNT_VPROP

Get postings with posting date and value type (EC)	**Definition** Select postings by posting date and value type **Description** This operation selects postings from the account, previously imported using the operation *Generate Account List*, by posting date and value type. The operation value is the value of the first selected posting. You can process the selected postings in a loop with *Next posting*. **Parameters** • Posting date from • Posting date to • Value type • All personnel numbers? If this indicator is set, all postings read from the database are considered for selection, regardless of personnel number. Usually, only the postings for the current personnel number are processed. **Prerequisites** An account list must have been generated. **Final Conditions** • 0 – Postings were selected • 1 – No corresponding postings exist **Exceptions** Postings not yet loaded	PKE_READ_ACCNT_BD

Get postings with posting date and value type attribute(ED)	**Definition** Select postings by posting date and value type **Description** This operation selects postings from the account, previously imported using the operation *Generate Account List*, by posting date and value type attribute. The operation value is the value of the first selected posting. You can process the selected postings in a loop with *Next posting*. **Parameters** • Posting date from • Posting date to • Value type attributes • All personnel numbers? If this indicator is set, all account postings read from the database are considered for selection, regardless of personnel number. Usually, only the postings for the current personnel number are processed. **Prerequisites** An account list must have been generated. **Final Conditions** • 0 – Postings were selected • 1 – No corresponding postings exist **Exceptions** Postings not yet loaded	PKE_READ_ACCNT_VP_BD

Get postings and totals with posting date and value type (EJ)	**Definition** Select postings by value date and value type. **Description** This operation selects postings from the account, previously imported using the operation *Generate Account List*, by value date and value type. The operation value is the total of the selected posting. **Parameters** • Value date from • Value date to • Value type • All personnel numbers? If this indicator is set, all account postings read from the database are considered for selection, regardless of personnel number. Usually, only the postings for the current personnel number are processed. **Prerequisites** An account list must have been generated. **Final Conditions** • 0 – Postings were selected • 1 – No corresponding postings exist. **Exception** Postings not yet loaded	PKE_SUM_ACCNT

Get postings with value date and value type attribute(EI)	**Definition** Select postings by value date and value type attribute. **Description** This operation selects and totals postings from the account, previously imported using the operation *Generate Account List*, by value date and value type attribute. (Selection similar to operation E5) The operation value is the total of the selected postings. **Parameters** • Value date from • Value date to • Wage type attribute • All personnel numbers? If this indicator is set, all postings read from the database are considered for selection, regardless of personnel number. Usually, only the postings for the current personnel number are processed. **Prerequisite** An account list must have been generated. **Final Conditions** • 0 – Postings were read • 1 – No corresponding postings exist **Exception** Postings not yet loaded	PKE_SUM_ACCNT_VPROP

Get postings with posting date and value type (EK)	**Definition** Select postings by posting date and value type **Description** This operation selects and totals postings from the account, previously imported using the operation *Generate Account List*, by posting date and value type. (Selection similar to operation EC). The operation value is the total of the selected postings. **Parameters** • Posting date from • Posting date to • Value type • All personnel numbers? If this indicator is set, all account postings read from the database are considered for selection, regardless of personnel number. Usually, only the postings for the current personnel number are processed. **Prerequisites** An account list must have been generated. **Final Conditions** 0 – Postings were selected 1 – No corresponding postings exist **Exception** Postings not yet loaded	PKE_SUM_ACCNT_BD

Get postings with posting date and value type attribute(EL)	**Definition** Select postings by posting date and VT attribute **Description** This operation selects and totals postings from the account, previously imported using the operation *Generate Account List*, by posting date and value type attribute. (Selection similar to operation ED). The operation value is the total of the selected postings. **Parameters** • Posting date from • Posting date to • Wage type attribute • All personnel numbers? If this indicator is set, all account postings read from the database are considered for selection, regardless of personnel number. Usually, only the postings for the current personnel number are processed. **Prerequisites** An account list must have been generated. **Final Conditions** • 0 – Postings were read • 1 – No corresponding postings exist **Exception** Postings not yet loaded	PKE_SUM_ACCNT_VP_BD

Next posting (E1)	**Definition** Get next posting from selection **Description** This operation reads the next posting, selected using the operation *Get postings with value type* or *Get postings with VT attribute,* and, if necessary, returns their value. **Parameters** None. **Prerequisites** You must have selected postings with one of the following operations: *Get posting with value type* or *Get posting with VT attribute*. **Final Conditions** • 0 – Posting retrieved • 1 – No further postings **Exceptions** None.	PKE_ACCNT_NEXT

Write posting (E2)	**Definition** Write posting to DB, if necessary as subsequent posting **Description** This operation writes a posting to a PF account. **Parameters** • Fund • Fund type The account type is used, together with the fund and the current personnel number to determine the account number. • Value type • Value • Posting date • Value date Indicator: Generate event If this indicator is set, an event PFBasicCH.EntryCreated is triggered, after entering the posting that starts a workflow, if the wage type has the *attribute 5*. Note that triggering an event requires a COMMIT WORK, and is therefore run time intensive. It is therefore better to use batch input for mass processing. • Indicator: Follow-on posting If this indicator is set, a posting is not set directly, but is returned to an internal table of the call up program. This enables you to generate the appropriate postings per dialog box with the calculated user default values. The operation *Lock account* writes all internal postings to the database as follow-on postings, whose value dates are prior to the new lock date. Therefore, you can enter account closure postings in a type of "atomic" operation, which minimizes the danger of inconsistencies. **Prerequisite** The user must have a write authorization for the respective accounts. To do this, use the operation Generate account list to import the account (if necessary, with Start = End), before postings are created. This then checks the account authorization. **Final conditions** 0 = o.k. **Exceptions** None.	PKE_WRITE_ENTRY

Attach to account list (E3)	**Definition** Load postings from DB account; if postings are loaded with authorization level *W* or *X*, the respective account is locked for the duration of processing **Description** This operation imports account postings from the database, where already existing postings are supplemented. Therefore, it is possible to summarize postings for several accounts. You can continue to process these postings using the operations: *Get posting with value type, Get posting with VT attribute* and *Next posting*. **Warning:** Before an insured person's postings are read, you must first initialize the account list using the operation *Delete account list*, so that no postings from a previously-processed personnel number are used. **Parameters** • Fund • Account type The account type enables you (along with the fund), to determine the account number. • Start of posting data • End of posting data • Authorization level This is the maximum account authorization needed for the current processing. If write authorizations are needed (Authorization level W or X), the respective account is locked for the duration of the processing. • Account selection All accounts are selected where the validity period overlaps parameters Posting start and End dates Final Conditions • 0 – At least 1 account found • 1 – No account found • 2 – User is not authorized • 3 – Account is locked (only checked for authorization level W or X) **Exceptions** None.	PKE_APP_ACCT_LIST

Delete account list (E4)	**Definition** Initialize postings **Description** Initializes the account list generated using operation *E3*. **Exceptions** None.	PKE_RFR_ACCT_LIST
Lock account (E6)	**Definition** Set a lock date for an account. **Description** This operation sets the lock date for an account. It is called up after successfully processing an account closure and prevents subsequent postings during this period prior to the lock date. If follow-on postings exist for the account to be locked, with a value date prior to the new lock date, these are written to the database. **Parameters** • Fund • Account type The account type enables you (along with the fund), to determine the account number. • Lock date **Prerequisites** The lock date cannot be before the latest lock date. **Final Conditions** • 0 = o.k. • 1 – Account is already locked **Exceptions** DB error while setting lock date	PKE_LOCK_ACCT

Test: Account locked? (E7)	**Definition** Test: Is account locked on this date? **Description** This operation checks if an account is locked on a specified date. In this way, you can skip the processing for those insured persons whose accounts have already been successfully closed. **Parameters** • Fund • Account type The account type enables you (along with the fund), to determine the account number. • Date **Final Conditions** 0 – Account is not locked 1 – Account is locked **Exceptions** None.	PKE_CHK_ACCT_LOCKED
Date of first posting **(E8)**	**Definition** Posting date of the first posting in the selection **Description** Result of the operation is the date of the 1st posting with a specified value type, as of a given date. If no posting exists, the date appears as 31.12.9999. **Parameters** • Value type • Date **Prerequisite** The account must have been imported using the operation *Generate account list*. **Final conditions** 0 = o.k. **Exceptions** Postings not yet loaded	PKE_FIRST_ENTRY

Date of last posting(E9)	**Definition** Posting date of the last posting in the selection **Description** Result of the operation is the date of the last posting with a specified value type, as of a given date. If no posting exists, the date appears as 01.01.1800. If no value type is specified, then all postings are taken into account. **Parameters** • Value type • Posting date **Prerequisite** The account must have been imported using the operation *Generate account list*. **Final conditions** 0 = o.k. **Exception** Postings not yet loaded	PKE_LAST_ENTRY
Value of last posting (EA)	**Definition** Value (by value date) of the last posting in the selection **Description** Result of the operation is the value of the last posting with a specified value type, as of a given value date. If no value type is specified, then all postings are taken into account. **Parameters** • Value type • Value date **Prerequisite** The account must have been imported using the operation *Generate account list*. **Final Conditions** • 0 = o.k. • 1 – No posting found **Exceptions** Postings not yet loaded	PKE_ACCNT_LAST

Lock date **(EB)**	**Definition** Account lock date **Description** Result of this operation is the lock date of an account. **Parameters** • Fund • Account type • Reference date **Final conditions** 0 = o.k. **Exceptions** None.	PKE_ACCT_LOCKDATE
Value date of current posting (EE)	**Definition** Value date of current posting **Description** Result of the operation is the value date of the current posting. **Final Conditions** • 0 – Posting exists • 1 – No current posting exists **Exceptions** None.	PKE_CUR_VADAT
Posting date of current posting **(EF)**	**Definition** Posting date of current posting **Description** Result of the operation is the posting date of the current posting. **Final Conditions** • 0 – Posting exists • 1 – No current posting exists **Exceptions** None.	PKE_CUR_BUDAT

Value of current posting (EG)	**Definition** Value of current posting **Description** Result of the operation is the value of the current posting. **Final Conditions** • 0 – Posting exists • 1 – No current posting exists **Exceptions** None.	PKE_CUR_VALUE
Dummy posting (EH)	**Definition** Create internal posting (with value type, posting date and value date) for simulations **Description** This operation generates a posting and writes it to the memory for simulation purposes. Follow-on operations that access postings can use the posting, as if they had been read from the database. **Parameters** • Value type • Value • Posting date • Value date **Final conditions** 0 = o.k. **Exceptions** None.	PKE_PSEUDO_ENTRY

Infotypes

Get infotype record (I0)	**Definition** Selects infotype records by subtype, validity period and object-ID, and loads the first of these records into the header of the corresponding infotype table. Records are selected from all infotype records that fall within the operation parameters. The first of these records is imported into the header of the infotype table, or – for time constraint 1 or 2 – the last (= latest) of these records. You can read these infotype fields to a value type with the operation *Read variable.* The next infotype in the selection is placed in the header using the operation *Next infotype record*. **Parameters** • Infotype (4-figure) • Subtype • Object-ID Generally not required. • Start • End **Prerequisites** The infotype table must have been transferred from the call up program. Note that the selection of data depends on the call up program. **Final Conditions** • 0 = o.k. • 1 – No record available for selection **Exceptions** Infotype not available.	PKE_IT_REC_GET

Next infotype record (I1)	**Definition** Places the next selected infotype record into the header. **Description** This operation places the next-selected infotype records (using operation I0) in the header of the infotype table. **Parameter** Infotype **Prerequisites** You must have defined a selection for the current infotype using the operation **Get infotype records.** **Final Conditions** • 0 = o.k. • 1 – No more records available **Exceptions** • Infotype not available • No records selected yet	PKE_IT_REC_NEXT
First field name for attribute class (I2)	**Definition** Provides the name of the first infotype 278 (P0278-EMAxy) field, that is assigned to the given attribute class. **Description** This operation places the next-selected infotype records (using operation I0) in the header of the infotype table. **Parameter** Infotype **Prerequisites** You must have defined a selection for the current infotype using the operation **Get infotype records.** **Final Conditions** • 0 = o.k. • 1 – No more records available **Exception** Attribute class is not assigned	PKE_FNAM_FRSTATT_278

First name of a free field for attribute class (I3)	**Definition** Provides the name of the first empty infotype 278 (P0278-EMAxy) field, that is assigned to the given attribute class. **Description** Result of the operation is the name of the first empty (relating to the current structure P0278) field in infotype 0278, that is assigned to the given attribute class. **Parameter** Attribute class **Prerequisites** The specified attribute class must be assigned to a field in infotype 0278, or the processing is terminated. **Final Conditions** • 0 = o.k. • 1 – No empty field found **Exception** Attribute class is not assigned	PKE_FNAM_FREEATT_278
Value type field name (I4)	**Definition** Specifies the name of the field in infotype 0279 (P0279-VAExy), that is assigned to the given value type. **Description** Result of the operation is the field name (calculated value), that is assigned to infotype 0279, relating to the current fund. **Parameter** Value type **Prerequisite** The value type must be assigned to a field in infotype 0279. **Final conditions** 0 = o.k. **Exception** Value type is not assigned	PKE_FNAM_VAL_279

Test: Is a class attribute available? (I5)	**Definition** Tests whether the given attribute is set in the current P0278. **Description** This operation checks whether the attribute in the attribute class is set in line with operation parameters in the current P0278, and sets the final conditions accordingly. **Parameters** • Attribute class • Attribute • **Prerequisites** **Final Conditions** 0 – Attribute exists 1 – Attribute not available **Exceptions** None.	PKE_CHK_ACLAS_278
Read value from infotype 0279 (I6)	**Definition** Reads a value from infotype 0279 using a given value type; if necessary, a manually-entered value. **Description** The value of this operation is the content of the field of the current structure P0279, that is assigned to the specified value type. If a manual value exists, it is selected. Otherwise, the calculated value is used. If the calculated value is to be read, use the operation *Read calculated value from infotype 0279*. **Parameter** Value type **Prerequisites** You must have filled the header of table P0279[] using the operation *Get infotype records*. Final conditions 0 = o.k. **Exception** Value type is not assigned	PKE_RD_IT_VAL_279

Write value in infotype 0279 (I7)	**Definition** Writes a value to the infotype 0279 (P0279-VAExy) field, that is assigned to the given value type. **Description** This operation writes the value of the current value type to the field (calculated value) of the structure P0279, that is assigned to this value type. **Prerequisites** You must have filled the header of table P0279[] using the operation *Get infotype records*. Final conditions 0 = o.k. **Exceptions** None.	PKE_WR_IT_VAL_279

Determine monthly pay from infotype 0008 **(I8)**	**Definition** Determines the current monthly pay from infotype 0008. The total of all specified wage types is formed, unless the wage types to be used are defined with the operation *Wage types for monthly pay*. **Description** Value of this operation is the monthly pay resulting from wage types in infotype 0008. Usually, an indirect valuation is carried out for each existing wage type; the total of all these wages, determined in this way, is then displayed. You can determine a subset of the wage types that form part of this calculation with the operation *Wage types for monthly wage*. For hourly-paid workers, this value is multiplied by the number of work hours per payroll period (field P0008-DIVGV), so that you always receive a monthly value. **Parameters** • Reference date This date is used to select an infotype 0001 record, and with which you determine the *Employee grouping for personnel calculation rule*, using the employee group and the employee subgroup. • **Prerequisites** The infotypes 0001, 0007 and 0008 must have been transferred to the PC interpreter on the reference date. **Final Conditions** • 0 = o.k. • 1 – Error during indirect valuation • 2 – No wage type found **Exceptions** • Arithmetical addition error • Arithmetical multiplication error (hourly wage → monthly wage)	PKE_RD_LGA_IT0008

Determine mean value from infotype 0279 (I9)	**Definition** The mean value of a value in infotype 0279 is determined for a given period. **Description** Value of this operation is the monthly pay resulting from wage types in infotype 0008. Usually, an indirect valuation is carried out for each existing wage type; the total of all these wages, determined in this way, is then displayed. You can determine a subset of the wage types that form part of this calculation with the operation *Wage types for monthly wage*. For hourly-paid workers, this value is multiplied by the number of work hours per payroll period (field P0008-DIVGV), so that you always receive a monthly value. **Parameters** • Reference date This date is used to select an infotype 0001 record, and with which you determine the *Employee grouping for personnel calculation rule*, using the employee group and the employee subgroup. • **Prerequisites** The infotypes 0001, 0007 and 0008 must have been transferred to the PC interpreter on the reference date. **Final Conditions** • 0 = o.k. • 1 – Error during indirect valuation • 2 – No wage type found **Exceptions** • Arithmetical addition error • Arithmetical multiplication error (hourly wage → monthly wage) **Exception** Value type is not assigned	PKE_RD_MEANVAL_279

Determine total value from infotype 0279 **(IA)**	**Definition** The total of a value is calculated in infotype 0279 for the payroll periods in a specified period. **Description** Result of this line is the total of the value for the specified value type using the specified time period. The calculation uses the number of employee's payroll area periods. Only those periods that lie completely within the specified period or validity period of infotype 0279 are counted. If there are insufficient payroll periods, a monthly basis is used. **Parameters** • Value type • Start of the period • End of the period **Prerequisites** • The specified value type must be assigned to a field in infotype 0279, for the current fund. • The infotype records that form the base for determining the mean value, must have been transferred from the call up program. **Final Conditions** • 0 = o.k. • 1 – No data in period: In the specified period, there is no infotype 279 record. **Exception** Value type is not assigned	PKE_RD_SUMVAL_279

Read calculated value from infotype 0279 **(IB)**	**Definition** Reads a calculated value from infotype 0279, using a specified value type. **Description** This operation reads the calculated value from the field in structure P0279, that is assigned to the specified value type in infotype 0279. However, operation I6 only reads the calculated value if no manual value available. **Parameter** Value type **Prerequisites** You must have filled the header of table P0279[] using the operation *Get infotype records*. Final conditions 0 = o.k. **Exception** Value type is not assigned	PKE_RD_IT_VAL_279
Determine wage types to calculate monthly pay from infotype 0008 **(IC)**	Definition Determines (up to 10) wage types that are to go into the monthly pay calculation. If this operation is not used, all wage types in infotype 0008 are used for the monthly pay calculation. **Description** This operation determines which wage types are used to determine the monthly wage, with the operation *Monthly wage from infotype 8*. If this operation is not used, all wage types in infotype 0008 are used for the monthly pay calculation. **Parameters** • Wage type 1 ... • Wage type 10 Final conditions 0 = o.k. **Exceptions** None.	PKE_LGA_SELECT

Determine mean value from infotype 0014 **(ID)**	**Definition** The mean value of the amounts in infotype 0014 records is determined for a given period. **Description** Result of this line is the mean value of the value for the specified value type using the specified time period. The mean value is formed using the number of employee's payroll area periods, relative to the total number of periods. Only periods that lie entirely within the respective validity period of infotype 14 are counted. If there are insufficient payroll periods, a monthly basis is used. **Parameters** • Wage type • Start of the period • End of the period **Prerequisites** • The specified wage type must exist in infotype 14. • The infotype records that form the base for determining the mean value, must have been transferred from the call up program. **Final Conditions** • 0 = o.k. • 1 – No data in period: In the specified period, there is no infotype 14 record. **Exceptions** None.	PKE_RD_MEANVAL_14

Wage type amount from infotype 0008 (IE)	**Definition** The wage type amount of the currently selected infotype 0008 record is read **Description** The value of this operation is the content of the field of the current structure P0008, that is assigned to the specified value type. **Parameter** Wage type **Prerequisites** You must have filled the header of table P0008[] using the operation *Get infotype records*. Final conditions 0 = o.k. **Exceptions** None.	PKE_RD_BETRG_8

PC Parameters

Preceding reference value (P0)	**Definition** Value of the first reference date from the preceding record. **Description** Result of this operation is the reference value 1 of the PC parameter, determined by the specified value type, so that the operation parameter reference value 1 lies between this value and the next. Example: The following value tables have been specified for the PC parameters (selection): Reference value 1 ... ... 1.00 ... ... 2.00 ... ... If the operation is called up with x : = reference value 1 (where the other key values can match correspondingly), the result of the operation = 1.00, if 1.00 <= x < 2.00. **Parameters** • Value type of the PC parameter • Gender • Reference value 1 • Reference value 2 • Reference date **Prerequisites** • The specified value type must be set up and maintained as a PC parameter (attributes, parameter area determination). • A suitable previous PC parameter record must exist. • exist. **Final conditions** 0 = o.k. **Exceptions** • Incorrect parameter area determination • Error accessing PC parameter	PKE_PAR_REF1_PREV

Subsequent reference value (P1)	**Definition** Value of the first reference value from the follow-on record. **Description** Result of this operation is the reference value 1 of the PC parameter, determined by the specified value type, so that the operation parameter reference value 1 lies between this value and the previous value. Example: The following value tables have been specified for the PC parameters (selection): ... Reference value 1 ... ... 1.00 ... ... 2.00 ... ... If the operation is called up with x := reference value 1 (where the other key values can match correspondingly), the result of the operation = 2.00, if 1.00 < x <= 2.00. **Parameters** • Value type of the PC parameter • Gender • Reference value 1 • Reference value 2 • Reference date **Prerequisites** • The specified value type must be set up and maintained as a PC parameter (attributes, parameter area determination). • A suitable follow-on PC parameter record must exist. **Final conditions** 0 = o.k. **Exceptions** • Incorrect parameter area determination • Error accessing PC parameter	PKE_PAR_REF1_ NEXT

Read PC parameter (P2)	**Definition** PC parameter value using given key value (must match exactly). **Description** Result of the operation is the value of the PC parameter with the specified value type that is defined as the specified key value of the operation parameters. **Parameters** • Value type • Gender • Reference value 1 • Reference value 2 • Reference date **Prerequisites** • The specified value type must be set up and maintained as a PC parameter (attributes, parameter area determination). • A PC parameter record must exist with the specified key values. **Final conditions** 0 = o.k. **Exceptions** • Incorrect parameter area determination • Error accessing PC parameter	PKE_RD_PAR

Read PC parameter with interpolation (P3)	**Definition** PC parameter value using given key value; if the reference value 1 does not match exactly, the value is determined by interpolating the previous and next value. **Description** A PC parameter is read with the key values specified as the operation parameters. If the reference value 1 cannot be matched, the value is determined by linear interpolation of the two "neighboring" parameter values. If there is only one "neighbor", this is used. **Parameters** • Value type • Gender • Reference value 1 • Reference value 2 • Reference date **Prerequisites** • The specified value type must be set up and maintained as a PC parameter (attributes, parameter area determination). • The PC parameters must have a numerical value. **Final Conditions** 0 = o.k. **Exceptions** • Incorrect parameter area determination • Error accessing PC parameter	PKE_RD_PAR_IPOL

Read previous PC parameter (P4)	**Definition** PC parameter value using given key value; if the reference value 1 does not match exactly, the previous value is read. **Description** Result of this operation is the value of the PC parameter, determined by the specified value type, so that the operation parameter reference value 1 lies between the corresponding record and reference value 1 of next record. Example: The following value tables have been specified for the PC parameters (selection): ... Reference value 1 ... Value ... 1.00 ... 3.00 ... 2.00 ... 4.00 ... ... If the operation is called up with x := reference value 3.00 (where the other key values can match correspondingly), the result of the operation = 1.00, if 1.00 <= x < 2.00. **Parameters** • Value type • Gender • Reference value 1 • Reference value 2 • Reference date **Prerequisites** • The specified value type must be set up and maintained as a PC parameter (attributes, parameter area determination). • A suitable previous PC parameter record must exist. **Final conditions** 0 = o.k. **Exceptions** • Incorrect parameter area determination • Error accessing PC parameter	PKE_RD_PAR_PRED1

Calculate retirement date PF (P5)	**Definition** Pension fund-dependent calculation of retirement date. **Description** The operation calculates the retirement date based on the pension fund from the view *Retirement Age Calculation* (V_T5CR0). The value type under which the result of the operation should be stored is specified as a subapplication key. **Parameters** • Subapplication key • Gender • Date of birth **Return value** **– Retirement date**	PKE_RD_PENSIONDATE
Calculate retirement age PF (P6)	**Definition** Pension fund-dependent calculation of retirement age. **Description** The operation calculates the retirement age based on the pension fund from the view *Retirement Age Calculation* (V_T5CR0). The value type under which the result of the operation should be stored is specified as a subapplication key. **Parameters** • Subapplication key • Gender • Date of birth **Return value** – Retirement age	PKE_RD_PENSIONAGE

Control

Test: Value1 = Value2? (S0)	**Definition** Test for same **description**. This operation tests two values for likeness and sets the corresponding final condition. **Parameters** • Value 1 • Value 2 **Prerequisites** Both values must be "compatible"; both numbers, or both the same. Note: In some cases it is necessary to force the category when entering a constant, using F4-Help. Example: Gender is not numerical. You must therefore enter a comparative value as a character string. **Final Conditions** • 0 – Value 1 = Value 2 • 1 - Value 1 <> Value 2 **Exceptions** Incompatible types	PKE_CONDITION_EQ

Test: Value1 = Value2? (S1)	**Definition** Test, whether a value is smaller or the same as a second value **Description** This operation tests whether a specific value is smaller or larger than a second value, and sets a corresponding final condition. **Parameters** • Value 1 • Value 2 **Prerequisites** Both values must be "compatible"; both numbers, or both the same. **Note**: In some cases it is necessary to force the category when entering a constant, using F4-Help. Example: Gender is not numerical. You must therefore enter a comparative value as a character string. **Final Conditions** • 0 – Value 1 <= Value 2 • 1 – Value 1 > Value 2 **Exceptions** Incompatible types	PKE_CONDITION_LE

Test: Value1 < Value2? (S2)	**Definition** Test, whether a value is smaller than a second value **Description** This operation tests whether a specific value is smaller than a second value, and sets a corresponding final condition. **Parameters** • Value 1 • Value 2 **Prerequisites** Both values must be "compatible"; both numbers, or both the same. **Note**: In some cases it is necessary to force the category when entering a constant, using F4-Help. Example: Gender is not numerical. You must therefore enter a comparative value as a character string. **Final Conditions** • 0 – Value 1 < Value 2 • 1 – Value 1 >= Value 2 Exceptions Incompatible types	PKE_CONDITION_LT

User operation (S3)	**Definition** User-Exit **Description** This operation allows you to enhance the available functionality of the processing schedule. To do this, call up the enhancement component '001' for the program SAPLRPCC. The User-Exit can be maintained using transaction CMOD (Enhancement PK020001 for development class PK02). It is located in function module EXIT_SAPLRPCC_001 of function group XPC1. A User Exit can be used to process other infotypes. **Parameters** • Enhancement activity The user exit uses this code to differentiate between processes (CASE Activity ...). • Transaction class This is used to determine whether employee ("A") or applicant data ("B") is read upon accessing infotype tables. If this field is empty, the value is used that was transferred to the interpreter. • Parameter 1 ... • Parameter 8 **Prerequisites** The enhancement must be maintained (coded in ABAP), and active. **Final conditions** The significance of the various final conditions depends on the implementation of the User Exit. • 0 = o.k. • 1 – Returncode 1 • 2 – Returncode 2 • 3 – Returncode 3 • 4 – Returncode 4 **Exceptions**	PKE_USER_EXIT

Test: Does the method exist? (S4)	**Definition** Using value type and method type, test whether a given PC method exists; this is especially useful prior to calling up an indirect method **Description** This operation tests whether a method exists with the current run time data (EE grouping, reference date). This can be used to prevent run time errors when indirectly calling up methods, where a default-processing is defined for non-existent methods. **Parameters** • Value type • Method type **Final Conditions** • 0 - Method exists • 1 - Method does not exist **Exception** Error evaluating the feature	PKE_TEST_METHD

Test: Does VT have attribute y? **(S5)**	**Definition** Test, whether a value type has a particular attribute. **Description** This operation tests whether a value type has a particular value attribute. **Parameters** • Value type • Value type attributes **Final Conditions** • 0 – Value type has attribute • 1 – Value type does not have the attribute **Exceptions** None.	PKE_TEST_VPROP
Day/Month **(S6)**	**Definition** Test: Day /Month in date. **Description** This operation tests whether a specified date contains a month and a day. This can determine, for example, whether a specified date is an end-of-year date. **Parameters** • Date • Month • Day **Final Conditions** • 0 - Day/Month is contained in date • 1 - Day/Month is not contained in date **Exceptions** None.	PKE_TEST_DATE
Subroutine complex **(SC)**	Not implemented.	

Subroutine (SD)	**Definition** Call up a submethod **Description** The processing of the current PC method is interrupted and a submethod is carried out. At the end of this submethod, the call up method continues. You can transfer up to 8 parameters when calling up a submethod. These values are assigned the pseudo-values types _P01 to _P08, after their previous values have been saved. In the submethod, _P01 to _P08 are replaced during the run time by operation parameter value type and value. If the submethod has finished, the previous values _P01 to _P08 are restored. This is a pseudo-operation (without code). The actual processing takes place in the interpreter. **Parameters** • Value type Usually (for a direct call up), indicate this **parameter** as a "constant". (Note capitalization!) • Method type • Parameter 1 Assigned pseudo-value type _P01. ... • Parameter 8 Assigned pseudo-value type _P08. **Prerequisites** The call up method must exist. This can be ascertained with the operation *Test: Does method exist?.* **Final conditions** The final condition is specified by the code that is specified in the final step of the submethod as a parameter of the final processing *Back*. • 0 – Returncode 0 • 1 – Returncode 1 ... • 4 – Returncode 4	

Various

Copy value (V0)	**Definition** Returns the value given as a parameter. **Description** Result of the operation is the value of the parameter. **Parameters** Value **Prerequisite** The data type of the value to be copied must be compatible with the target value type (current value type). **Final conditions** 0 = o.k. **Exceptions** Incompatible data types	PKE_COPY
Error message (V1)	**Definition** Adds an error message to the current error list; it appears, for example, in the error list of a report log. **Description** The operation adds an error message to the error list. This is then available for analysis, for example, after processing a universal report. **Parameters** • Work area • Message category • Message number • Message parameter 1 • Message parameter 2 • Message parameter 3 • Message parameter 4 **Final conditions** 0 = o.k. **Exceptions** None.	PKE_ERROR

Evaluate feature (V2)	**Definition** Provides the value of the current feature, which results from the current employee data in infotype 0001 **Description** This operation evaluates a feature with the current data from infotype 1. The result is the value of the feature. **Parameter** Name of the feature **Prerequisites** The feature must use the structure PME17 and return a simple value that is no longer than 18 characters. **Final conditions** 0 = o.k. **Exception** Error evaluating the feature	PKE_EVAL_FEATURE
Read program variable (V3)	**Definition** Provides the value of the given program variables. **Description** Result of the operation is the content of the specified program variables. Parameters Name of the program variables The name must be known. Usually, this operation is required to access infotype fields. You can find out the names of the fields using the data dictionary. Example: Date of birth is read from P0002-GBDAT. **Prerequisites** The variable must exist in the call up program (that is, SAPLRPCC or SAPMPACC). Otherwise, the program is cancelled. **Final conditions** 0 = o.k. **Exceptions** • Variable does not exist. • For infotype fields: Infotype has not been retrieved (Operation I0).	PKE_GET_VAR

Read payroll result (V4)	**Definition** Reads the payroll result for a period and transfers the wage type amount to the RT. **Description** This operation imports a current payroll result (Indicator 'A'), for the current employee and reads the entry in the header of the payroll table RT[] with the specified wage type. You can transfer the corresponding value to a value type with the operation *Read variable.* If the operation is repeated for the same payroll period, but a different wage type is called up, the import is not repeated, as long as the payroll result is not overwritten by a new personnel number or a different payroll period. **Parameters** • Payroll periods (for) • Payroll year (for) • Wage type **Final Conditions** • 0 = o.k. • 1 – Error during import • 2 – Wage type is not in RT **Exceptions** None.	PKE_READ_PAYROL L
Interruption (VT list) (V5)	**Definition** Lists values in current processing **Description** This operation generates a list specifying the current position in the processing schedule and a list of the current values. You can use this for debugging. Parameters • Text 1 • Text 2 Use the text lines to more easily differentiate between interruptions. **Final conditions** 0 = o.k. **Exceptions** None. (The operation forces an exceptional processing.)	PKE_BREAK_POINT

Test: Is value 0? (V6)	**Definition** Test, whether the current value type value is 0 (type-dependent, = SPACE) **Description** This operation tests whether the current value type is 0 (= SPACE), and sets a corresponding final condition. **Final Conditions** • 0 – Value is not zero • 1 - Value is zero **Exception** Value type not defined	PKE_TEST_INITIAL
Guaranteed payment (V7)	**Definition** Guaranteed payment maintenance for insured wage **Description** The result of this operation is the employee's new insured wage, considering a changed coordination deduction or capacity utilization level. If the new insured wage is not lower than the old one, the first result is displayed. This is also the case if the coordination deduction was reduced, or if the capacity utilization level is to change. Otherwise, the previous insured wage is returned as the result. **Parameters** • Previous insured wage • New insured wage (in line with calculation) • Previous coordination deduction • New coordination deduction • Previous capacity utilization level • New capacity utilization level **Final conditions** 0 = o.k. **Exceptions** None.	PKE_BESITZ

Number of payroll periods **(V8)**	**Definition** Number of payroll periods in one period. **Description** Result of the operation is the number of payroll periods in the specified period. This is determined using the period modifier of the last payroll area, to which the employee is assigned (using infotype 1), in the period. **Parameters** • Start of the period • End of the period **Prerequisite** The payroll periods must be generated for the period modifier in the specified period. **Final conditions** 0 = o.k. **Exceptions** None.	PKE_NUM_OF_PER
Save values **(V9)**	**Definition** Save up to five values under one key in the value buffer. **Description** Write up to 5 values with key in the value buffer. If values already exist as specified operation parameter value types, they are overwritten with the current ones. By repeated call up, you can save as many values as you wish. These values are saved, even if you process several personnel numbers. **Parameters** • Key • Value type 1 • Value type 2 • Value type 3 • Value type 4 • Value type 5 **Final conditions** 0 = o.k. **Exceptions** None.	PKE_SAVE_VALS

Restore values (VA)	**Definition** Restore values from the value buffer according to the key. **Description** The current values that are saved in the value buffer, under the key, are restored. If you enter '*' as the key, all postings are restored. You retain the saved values. **Parameter** Key **Final conditions** 0 = o.k. **Exceptions** None.	PKE_REST_VALS
Delete value buffer (VB)	**Definition** Delete value in value buffer with given key. **Description** The current values that are saved in the value buffer, under the key, are deleted If you enter '*' as the key, all postings are deleted. **Parameter** Key **Final conditions** 0 = o.k. **Exceptions** None.	PKE_CLEAR_BUF

Processing Models

Use

A *Processing model* is an interplay of certain groups of operations.

An example of this would be that individual operations create global data, that are required for other operations.

See also:

Processing Model: Infotypes [Page 213]

Processing Model: Accounts [Page 214]

Processing Model: Value Buffer [Page 216]

Processing Model: Follow-on Postings [Page 217]

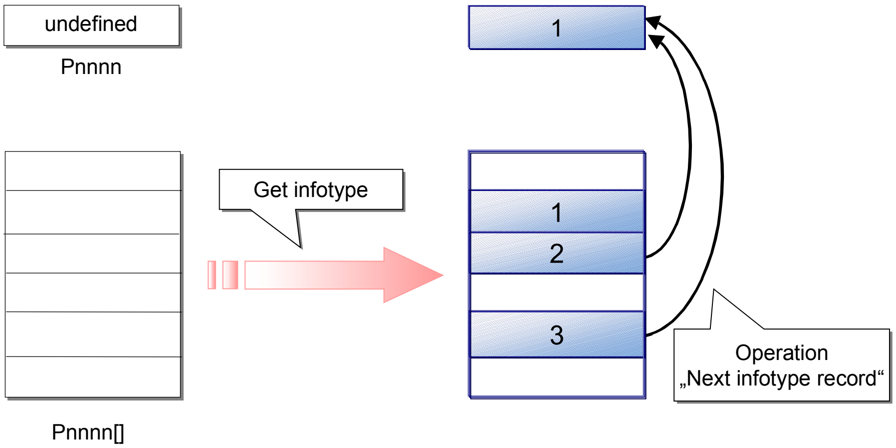
Processing Model: LOB [Page 218]

Processing Model: Infotypes

Use

When processing infotypes, note that they are transferred from the call-up program to the Interpreter as an internal table. If you select particular records in a PNP report (for example, per selection period), they can only be used in processing control.

Getting an infotype record is the same as selecting records by certain criteria, and importing this record into the header of the infotype:



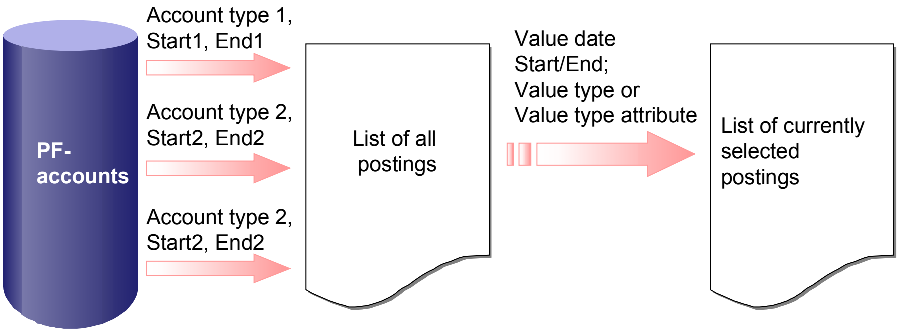
When selecting infotypes, their time constraint is taken into account: if only one record can exist at one time, the newest record is chosen from the selection period.

Processing Model: Accounts

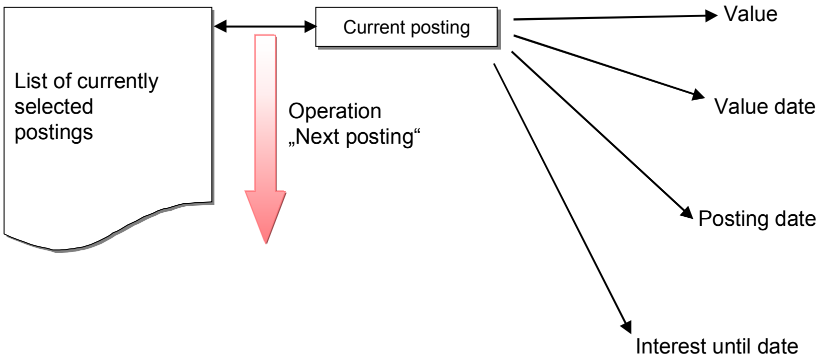
Use

Pension fund accounts can be accessed on two levels:

- Create an account list: the account postings are imported to an internal table; you must specify the account type and period (for posting data)
- Select postings according to value type or value type attribute; the first selected posting is read as the operation result
- Read other selected postings


PF-accounts
Account type 1, Start1, End1
Account type 2, Start2, End2
Account type 2, Start2, End2
List of all postings
Value date Start/End; Value type or Value type attribute
List of currently selected postings

From this selection, all postings can be processed in a loop, where the operation *Next posting* always selects the respective posting value. You can determine the value date and the posting date, or interest with further operations:



Reading a **savings credit** (Cumulation of postings)
When you read a savings credit, the postings are selected, as long as the account has been read by the database. The result is the value of the first posting. If no posting exists, do not report an error. The return code is 0.
In a second step, the Interpreter goes over the other postings in a loop; the value of the current posting is cumulated to the previous value by the line type *SUM*. When the loop is finished, the return code should be 0.

Processing Model: Value Buffer

Definition

To save temporary values, use a *value buffer*. You can do this with three operations:

- Save values
- Restore values
- Delete value buffer

The values are stored with a key term, so that more buffers are available.

Use

When you start to process a personnel number, the buffer is not initialized. Therefore, it is possible to use data from an old personnel number. For example, use the current personnel number as the key.

Processing Model: Follow-on Postings

Definition

Under certain conditions, a certain calculation must be followed by the creation of a certain posting.

- Creating an *EHO Advance* posting → *LOB Advance* posting (according to fund)
- Benefit plan entry calculation: Posting of benefit reduction at age of entry (compensated by LMB)
- Benefit-plan capacity utilization level change: Posting of benefit increase (reduction in capacity utilization level) or benefit reduction (benefit increase)
- Benefit plan premiums: corresponding benefit increase

Use

This is to be implemented so that when you call up the PC operation *Create posting,* an indicator *As follow-on posting* is specified. The postings are then transferred to the call up program (currently a manual posting maintenance, user-interface [Function *Create infotype 0279*], method PFValsCH.VSAction), where you can process the individual postings in a dialog window with the function *Change*.

You can directly write the postings from the workflow-inbox or the PF-user-interface. At this point, a different interface status is defined.

Processing Model: LOB

Use

LOB calculations/processes are contained, as far as possible, in one operation.

The group of operations to determine the minimum leaving benefit, according to Art. 17 of the FLV, are special: The actual calculation only takes place once; the various operations are used to read the various results.

Design of PC Activities

The design of PC activities is largely determined by the type of PC Interpreter call up that you use to call up the activity.

The interpreter is called up from:

- Universal Evaluations
- Online Evaluation
- The Method PFValsCH.VSAction
- The Function Module HR_CH_PK_WRITE_ENTRY_POPUP_KNR, for creating postings manually
- The Module Pool MP027900

PC Activities for Universal Evaluations

The following parameters are transferred to the Interpreter:

Reference date	Selection Screen
Start	PNPBEGDA
End	PNPENDDA
Fund	Current Fund (Loop using infotype 278 for each personnel number)
PC Activity	Dependent on PC schema
Personnel number	Current personnel number
Transaction class	*B* for RPLPKBC0, otherwise *A*
Error handling	Dialog: Yes, Background: No
Test mode	Selection Screen
Origin	Activity
Parameter table	Additional selection parameters
Infotypes	0, 1, 2, 7, 8, 14, 15, 36, 278, 279; Selection in PNP

PC Activities for Online Evaluations

The following parameters are transferred to the Interpreter:

Reference date	Dialog-box entry-field
Start	Default: Start of the reference-date year*
End	Default: End of the reference-date year*
Fund	Current fund (reference date-dependent)
PC Activity	According to the Customizing for the current calculation
Personnel number	Current personnel number (context-dependent)
Transaction class	A
Error handling	Yes
Test mode	Yes
Origin	OR (not relevant due to test mode)
Parameter table	– (used later for additional parameters)
Infotypes	0, 1, 2, 7, 8, 14, 15, 36, 278, 279; read between 01.01.1800 and 31.12.9999

* can be overridden per dialog box.

PC Activities in Workflow

PC Interpreter Call Parameters:

Reference date	WF Data flow
Start	WF Data flow
End	WF Data flow
Fund	WF Data flow
PC Activity	WF Data flow
Personnel number	Personnel number of the object with category PFValsCH
Transaction class	A
Error handling	No
Test mode	No
Origin	WF
Parameter table	WF Data flow (Value table)
Infotypes	1, 2, 7, 8, 14, 15, 36, 278, 279; read between start and end

PC Activities for Manual Postings

You have two interpreter calls for creating manual postings: to determine the value date and possible follow-on postings.

See also:

- Manual Posting: Determine Value Date (02) [Page 224]
- Manual Postings: Follow-on Posting (22) [Page 225]

Manual Posting: Determine Value Date (02)

Purpose

This activity uses the possibility of transferring value types as parameters to the PC interpreter. These are the value types VT00 (contains posting value type), and DTBU (contains posting date).
The system returns the value date in the value type DTVA.

This indirectly calls the *calculate VT00 value date* method. For example, if VT00 contains the value type ABCD, the *calculate value date 'ABCD'* method is called.

PC Interpreter Call Parameters:

Reference date	Posting date
Start	01.01.1800
End	31.12.9999
Fund	Account fund
PC Activity	02
Personnel number	Account holder's personnel number
Transaction class	'A'
Error handling	Yes
Test mode	No
Origin	–
Parameter table	VT00 with posting value type DTBU with posting date
Infotypes	None

Manual Postings: Follow-on Posting (22)

Purpose

This activity is structured similarly to the determination of the value date. You determine whether you need to create a follow-on posting for the current posting, and if so, the value of this posting.

The entry parameters are the same as for activity 02. The follow-on postings created using the PC operation *Create posting* (E2) are returned to an internal table.

The calling program (postings maintenance) then ensures that the determined additional postings are created.

PC Interpreter Call Parameters:

Reference date	Posting date
Start	01.01.1800
End	31.12.9999
Fund	Account fund
PC Activity	22
Personnel number	Account holder’s personnel number
Transaction class	‘A’
Error handling	Yes
Test mode	No
Origin	–
Parameter table	VT00 with posting value type DTBU with posting date DTVA with value date
Infotypes	1, 2, 8, 278, 279; read on the posting date

Processes in the *Individual Values PF* Infotype (0279)

You can assign PC activities to fields in the *Individual Values PF* infotype (0279) that are run when you make manual entries into the respective field. For example, activities “03” and “04” in the standard schedules.

If necessary, you must explicitly rewrite calculated values in the infotype record. Note that you **cannot** reset the original value (for example definitive wage), as it has been marked as a calculated value.

PC Interpreter Call Parameters:

Reference date	Infotype validity start date
Start	01.01.1800
End	31.12.9999
Fund	Current subtype
PC Activity	The PC activity assigned to the current field
Personnel number	Personnel number for the infotype-record
Transaction class	‘A’
Error handling	Yes
Test mode	No
Origin	–
Parameter table	Current value table
Infotypes	1, 2, 8, 278, 279; read at the start of the validity period for the infotype

Error Handling

Fehlersuche: Konsistenzprüfung im VS-Editor [Page 229]

Fehlersuche: PK-Debugger [Page 230]

Fehlersuche: Unterbrechungen [Page 231]

Purpose

Within processing control, there are several ways in which the system reacts to errors during the run time:

- Error message, or a list of error messages
- Exceptional processing
- In some cases, the ABAP program may terminate. To avoid these cases, run the PC editor consistency check.

Prerequisites

The following errors result in exceptional processing within the interpreter:

- **Missing Activity Assignment**: The activity that starts the PC interpreter is not assigned to a PC method.
- **Missing Method Variant**: No method variant exists for the called-up PC method as of the processing reference date. Probable cause: Called-up method does not exist.
- **Undefined Value**: No value exists in the current processing for an operation parameter-assigned value type.
- **Undefined Value Type**: A value type has been used that is not defined in the current processing context. This error occurs when you call up a method for the respective value type, as a value type must be assigned to a value (if necessary – blank), by one of its assigned methods, before it can be used.
- **Error in Feature Evaluation:** An error occurred when evaluating the feature to determine the EE grouping for the PC method determination. Probable cause: The feature is not active.
- **A Line is Missing in the Processing Schedule:** You have specified a non-existent line as the next processing step to be called up within a PC method (not a submethod call).
- **Return is Missing**: You have left a PC method, where you defined the last processing step as the “next” follow-on step. Here, you must have carried out a return “Back...”.
- **Unknown Operation**: You have defined an operation for the current processing step that is not defined in the existing processing context.
- **Error in PC Operations**: These exceptions are specified in the list of processing context operations for the *Swiss Pension Fund*.

Procedure

IF “minor errors”/warnings are present (for example, inconsistent or missing infotypes), which were caused by the user, or which can be resolved, you will see the following symbol in various programs (evaluations, online-evaluation) . Use this button to call up a dialog box that displays the error list. Select (double-click) on the relevant line to call up a more detailed error description with correction notes.

The system then runs an exceptional processing if the current problem cannot be logically resolved within the processing schedule. Such errors do not occur if the system has been set up correctly. This type of error cannot normally be corrected by the user.

During an exceptional processing, a dialog box opens, where you can see a summary and a detailed description of the processing, the exact location, and a list of already-calculated values.

A problem can arise if the PC interpreter is called up from a background task within a workflow. Therefore, the method PFVALSCH.VSACTION sends the exceptional processing task or error to the specified system user, per mail.

For the object method, this is controlled by the parameters *Recipient* (of an error list), *PC_Admin* (Recipient of exceptional processing results) and *WF_ID* (to display the WF-ID).

Troubleshooting: PC Editor Consistency Check

Purpose

The PC editor can detect errors or possible errors by calling up the *consistency check* function.

It checks the following:

- The validity of operation parameters by data type and PC value set
- The existence of call up methods (an error message appears for indirect call-ups, as the existence of the called method is only detected with the run-time data)
- The existence of blank processing lines (Operation ‘NO’)
- The existence of value types that not defined as operation parameters in the schedule (Warning: *undefined parameters*)
- The existence of unavailable lines
- Missing returns from methods

Warning messages (for example, *Indirect method call up*) indicate possible errors or causes of errors that should be checked.

Troubleshooting: PF Debugger

Use

Some errors cannot be detected in the PC Editor. This is especially the case for logical errors. To detect these, use the debugger.

Features

The debugger has the following functions:

- It imports employee or applicant master data
- It executes PC activities entirely, step-by-step, or per method
- It sets entry parameters
- It sets the reference date, start and end of the period
- It displays a log and the current value table

When you reset, the corresponding processing schedule and the infotype records are re-imported from the database.

> Note that a subprogram appears *twice* in the log: firstly, the actual jump and then the return; both are specified with the key from the call up processing step.

Using the details in a processing step, you receive the parameter values that call up the operation, and the operation result (value and return code)

Certain PC activities require entry parameters, for example, 02, 22, or entry simulations. Use the *Copy parameter* function to copy parameters to the value table. Use the *Parameter table* to default the entry parameters using the current PC activity.

Troubleshooting: Interruptions

Purpose

To locate logical errors, you can define interruptions in the processing schedule (operation V5). If the PC interpreter detects such an operation, it displays the call-up location and a list of currently-calculated value types.

Documentation Concept

The following objects can be documented within processing control:

- Value types (schedule-independent, as definition)
- Value types (schedule-dependent, as documentation for the implemented methods)
- PC schedules
- PC Schemas
- Online evaluation calculations
- PC operations (not relevant for customers)

The documents are created as standard texts (Documentation object *DOKU*, Text-ID *TX*); the assignment of the object to be documented takes place using the naming convention. This is automatically taken into account when maintaining the documentation texts with the transaction PAC0.

A SAP and a customer version can exist for each documentation text. If necessary, the customer version is used.

Naming convention for documents:

HRCH_VSDOC_<DOCUTYPE>_<CONTEXT >_<PPLAN><OBJNAME>

and for customer documentation:

ZHRCH_VSDOC_<DOCUTYPE>_<CONTEXT >_<PPLAN><OBJNAME>,

where the *<DOCUTYPE> is:*

- *'DF'* for value type definitions,
- *'IM'* for implementation notes for value types,
- *'PL'* for processing schedule documentation,
- *'SC'* for PC schema documentation,
- *'CL'* for calculation documentation,
- *'OP'* for PC schema documentation.

.

<PPLAN> is the name of the corresponding PC schedule for schedule-dependent documents (previously: Implementation notes for value types), or it is unnamed.

<CONTEXT> is the respective PC context: that is, *02_PF* for the Swiss pension fund.

User Exits

Definition

The number of standard operations should be enough for most requirements. In addition, there are user exits, especially for the following cases:

- If an implementation with existing operations is too difficult, or low-performance.
- If you have to process other infotypes (for example, customer-specific infotypes).

Use

User exits are implemented with the operation *User Operation* for each function module call up. This supports the SMOD/CMOD enhancement concept.

To use this, you must process the enhancement *PK020001* in the development class PK02 on the customer page, using the transaction CMOD.

You must code the user exit in the function module “EXIT_SAPLRPCC_001”. Use the transaction *CMOD* to see a sample coding.

The following parameters are available in the function module:

ACTION	Controls processing within the function module.
TCLAS	Transaction class: “A” for employee, “B” for applicant
PAR_RECORD	Parameters used to call up the PC operation. This triggers value types. The structure has the following fields: PAR_RECORD -PAR01 ... -PAR01 -VTY01 ... -VTY01
EVL	This structure is used to control calculation. It contains the following fields: EVL -CUR_VAL -CUR_VALTY -PRV_VAL -PRV_VALTY -SUBRC

VALUES	Current values table: VALUES -VALTY -GLBFL -VALUE
FUND	Current fund.
PERNR	Current personnel number

Implementation Tools

Löschen von PK-Konten [Page 240]

An specific IMG structure (SIMG_OHPK) has been set up for the Swiss pension fund. When creating an implementation project, specify the components PA-PF-CH (Personnel Management → Pension Plan → Pension Fund).

The following information is available for the SAP system context:

- Implementation Guide (IMG) [Page 236]
- Entity Copier [Page 237]
- Template Copier for Pension Funds [Page 238]
- Report: Creating PF Accounts [Page 239]

Implementation Guide (IMG)

A specific IMG structure (*SIMG_OHPK*) has been set up for the Swiss pension fund. This structure contains all steps that are required to implement and adjust the pension fund module.

When creating an implementation project, specify the components PA-PF-CH (Personnel Management –> Pension Plan –> Pension Fund).

Entity Copier

To copy (or to process differently), all dependent entries from an entity to a table, use the entity copier (function group *ECOP*). You control this using entries in tables *ENTCOPY00* and *ENTCOPY02*, and with user-exits.

For pension funds, use this for funds, processing schedules, parameter areas and parameter groupings. You must take into account the relevant processing context for processing control entities. Use the function module *HR_CH_SET_VS_CONTEXT* as a user-exit.

Template Copier for Pension Funds

The template copier for pension funds (RPUKASC0) is a tool to copy a model fund and dependent entities in their entirety to a new fund. Proceed as follows:

1. Copy the processing schedule, parameter range determination, main parameter range and the fund
2. Transfer the PC assignment for the new fund
3. Maintain the parameter range determination

The status of each step is shown by a light on the right side of the group box.

Enter the key for the new processing schedule, PaGru and main parameter range in the selection screen. This avoids foreign key problems.

Creating PF Accounts (RPUACOC0)

Use

The report *RPUACOC0* creates individual accounts for a headcount according to the specifications on the selection screen. In this way, you can set up test cases.

Deleting PF Accounts (RPUACOC1)

Use

The report RPUACOC1 deletes postings from individual accounts and complete accounts. The report cannot be run in a productive system.

Legacy Data Transfer

Musterreport für Umsetzung "PK-Alt" [Page 242]

Musterreport für Übernahme von Buchungen aus Altsystemen [Page 243]

Musterreport für Übernahme von Stammdaten aus Altsystemen [Page 244]

Features

There are no programs to transfer data from an old system, only sample reports. You must modify these to your requirements. The main task of the report is to create PF accounts and postings.

Get details regarding the reports from the corresponding report documentation.

The areas where modifications are required are indicated with *(K)*.

Sample Report for Converting "Old PF"

Use

The report *RPUPKUC0* can create a batch-input session for converting master data, as well as creating new accounts and transferring postings from cluster BV.

Sample Report for Transferring Postings from the Old System

Use

Report *RPUADUC0* imports a text file (1 line ↔ 1 posting) for the following structure from the workstation for each fund:

Field	Format	Meaning
PERNR	NUMC8	HR personnel number
BUDAT	YYYYMMDD	Posting date
VADAT	YYYYMMDD	Value date
VALTY	CHAR4	Value type
VALUE	CHAR12	Value

This file is loaded per WS_UPLOAD, and therefore must be present on the workstation, where the respective session GUI is running.

The earliest posting date of a posting to this account, and the entry parameter *Validity Start Date for Accounts* is formed as the start date for each account, and the relevant start of the month is calculated.

The sample report cannot be run. That is, for security reasons, it is ended at the “INITIALIZATION” time point. Therefore, to use it, it must first be modified.

Sample Report for Transferring Master Data from Old Systems

The report *RPIADUC0* creates a batch-input session for PF master data. The necessary data is taken from text files on the workstation.

One of the description files on the selection screen determines:

- Which data is imported
- Which infotypes these files are assigned to
- Certain constant field contents (for example, BEGDA or ENDDA)
- The assignment of data file columns to screen fields of the infotype module pool
- The assignment of data file columns to value types (→ Infotype 0279)

The description file has the following elements:

```
#INFOTYPE <Infotype number>       <Data file name> [<BDC_OKCODE>]
#FIELD    <Screen field name> #VAL <Constant value>
#FIELD    <Screen field name> #VAL <Column number>     [<Column width>]
#VALTY    <Value type>              <Column number>      [<Column
width>]
```

You can define how many #FIELD or #VALTY lines exist for each infotype. (The #VALTY is only relevant for infotype 0279.) The file can have several blocks with #INFOTYPE.

Note that you must write the key words in capitals.

The data file lines have the following structure:

PERNR [NUMC(8)]	SUBTY [CHAR(4)]	LINE [CHAR(1024)]

Run Time Aspects

DB Indices

In payroll, you do not have to access postings purely by using primary keys. For this reason, a secondary index PAY for table T5CPB has been created that contains the KONNR, PABRP and PABRJ fields. This index should be active.

Other indices are not required and can limit performance.

Memory Requirement for Evaluations

A PC universal evaluation data record (lines in table PR_TAB[]) is approximately 1.5 Kbytes. Therefore, to evaluate 1,000 employees (for 1 fund), you need approximately 1.5 MB on the application server.

Directory of Objects for PF/PC

Programme

Programs

MP027800	Module pool for the *Basic Data PF* infotype (0278)
MP027900	Module pool for the *Individual Values PF* Infotype (0279)
RPIPKLC0	PC Universal evaluation list, BI (obsolete)
RPIADUC0	Transfer of master data (Batch-Input)
RPLPKAC0	PC Universal evaluation SAPscript (obsolete)
RPLPKBC0	PC Applicant evaluation SAPscript
RPLPKKC0	Account Evaluation
RPLPKPC0	Complex person selection
RPLPKSC0	Universal Evaluation
RPLPKTC0	PC Universal evaluation list (obsolete)
RPU45BC0	Converting the initial values in domain P02_ACTIO (Rel. 4.5B)
RPU46AC0	Conversion for creating function views in the user-interface (Rel. 4.6A)
RPUA79C0	Reorganizing infotype 0279 (as of Release 4.X)
RPUACOC0	Report: Create employee accounts
RPUACOC1	Report: Delete accounts and postings
RPUADUC0	Model report: legacy data transfer (postings)
RPUKASC0	Pension funds-total copier
RPUPENC0	Postings according to payroll run
RPUPKCC0	Call up IMG (Transactions PUC*)
RPUPKFC0	Subroutines for external Performs from dynamic actions (PK recalculation, Entry, Create account)
RPUPKSC0	Conversion for new universal evaluation 4.6A
RPUPKUC0	Model report: Conversion “Old PF → New PF”
RPUVALC0	Carry out PC activity for workflow
RPUVSEC1	PC Editor
SAPLRPCC	Function group RPCC: PC interpreter for processing context “PF Switzerland”
SAPLRPCH	Function group RPCH: Online-Help
SAPLRPCP	Function group RPCP: Accounts, Postings
SAPMPACA	User Interface
SAPMPACB	Account Management
SAPMPACC	PF Debugger

SAPMPACE	*Classic* posting maintenance
SAPMPACO	Posting maintenance
SAPMPACT	PC parameter maintenance

Function Groups

RPCC	Interfaces for processing context Swiss pension fund
RPCV	PC Functions
RPCH	F1-/F4-Help
RPCP	PF Accounts
RPCR	PC Universal Evaluations

Transactions

PAC0	PC Administration
PACA	User Interface
PACB	Account Maintenance
PACE	*Classic* posting maintenance
PACEN/PACO	Posting maintenance
PACS	Applicant evaluation (Entry simulation)
PACN	Number range management for PF accounts (Number range HRCHPKONTO)
PACC	PF Debugger
PACT	PC parameter maintenance
PUCA	PC Administration “PF Switzerland” (for IMG)
PUCF	PC Form maintenance (for IMG)
PUCG	Template Copier for Pension Funds (for IMG)
PUCK	Entity Copier for Pension Funds (for IMG)
PUCV	Entity copier (for IMG; Domain over Subobject)
PUCP	PC parameter maintenance (for IMG; Subobject = ‘PAFGR’ → Selection with PaGru, or by fund)
PUCE	PC Editor (for IMG)
PUCW	Call up task maintenance (WF; for IMG; Subobject contains name of task)

Tables

Table	Class	Class namespace	Description
T5CP0/1	E	Fund #+++	Pension funds
T5CP2	C		PF → Processing Control
T5CP3	E	Fund: → T5CP0	Field assignment infotype 0279
T5CP4	C		Infotype fields for BI
T5CP5	C		Infotypes for BI
T5CP6/7	G	PFGrp: 8+, 9+	EE groupings for field assignment infotype 0279
T5CP8	C		Outline text for infotype 0279
T5CP9	C		Infotype for overview of user-interface
T5CPA	E	Fund: → T5CP0	Value types → Wage types
T5CPB	A		Postings
T5CPC/D	E	Calc: X+, Y+, Z+	Calculations for infotype 0279
T5CPE	E	Fund: → T5CP0	Control of IT 0279 calculations
T5CPF/FT	E	Fund: → T5CP0	Function view of a fund
T5CPG/H	C		Authorization groups for PF accounts
T5CPI	G	Schem: → T5CB4	Pension fund PC schema settings
T5CPJ/JT	E	Fund: → T5CP0	Fund function view categories
T5CPK	A		Account name
T5CPL	G	PFGrp: → T5CP6	Column headers for infotype 0279 list
T5CPM	G	Fund: → T5CP0	Columns for infotype 0279 list
T5CPO/P	E	Ipset	Entry parameter amount for infotype 0279 calculations
T5CPQ	C		Definition of entry parameter amounts
T5CPR	E	Fund: → T5CP0	Additional infotypes for PF recalculation
T5CPS	E	Fund: → T5CP0	Additional infotype field values for PF recalculation
T5CPT/TT			not released
T5CPU/UT	E	PfFunction: #+	PF Functions
T5CPV	E	Fund: → T5CP0	Control of PF functions
T5CPW	C		Field assignment query
T5CPX			not released
T5CPY/YT	E	Acvie: #+	Views for individual PF accounts

T5CPZ	A		Account numbers ↔ Personnel numbers
T5CQ0	E	Pplan: → T5CA4	Account view control
T5CQ1	E	Pplan: → T5CA4	Posting value types of an account view
T5CQ2	E	Pplan: → T5CA4	Order of the account views
T5CQ3	E	Pplan: → T5CA4	Resulting value types from an account view
T5CA0/1	E	Aclas: → T5CAP	EE attributes
T5CA2/3	E	Valty: #+++, _#++	Value types
T5CA4/5	E	PPlan: #+	Schedules
T5CA6/7	E	Methd: 9+	Method types
T5CA8	E	Pplan: → T5CA4	Activity assignment
T5CA9	E	Pplan: → T5CA4	Method variants
T5CAA/B	E	Parar: #+	Parameter areas
T5CAC	E	PaFGr: → T5CAW	Parameter area determination
T5CAD	E	Parar: → T5CAA	PC Parameter
T5CAE/F	S		PC Operations
T5CAG	E	Pplan: → T5CA4	Processing schedule: Method definitions
T5CAH	E	Pplan: → T5CA4	Schedule Value Types
T5CAI/J	E	EGMtd: #+ Pplan: → T5CA4	EE Grouping for Method determination
T5CAK	S		Text template for operation requirements
T5CAL	S		Operation parameter information
T5CAM	S		Operation parameter texts
T5CAN	S		Text templates for line types
T5CAO	E	Pplan: → T5CA4	Line markers for processing schedule
T5CAP/Q	E	Aclas: #+	EE attribute classes
T5CAR	C		EE attribute compatibility
T5CAS	C		Field assignment for EE attributes
T5CAT/U	E	VProp: #+	Value Type Characteristics
T5CAV	E	VProp: → T5CAT	WT characteristic assignment
T5CAW/X	E	PaFGr: #+	Parameter area determination groupings
T5CAY/Z	E	Actio: {6,...,9}+	PC Activities
T5CB0/1	E	VaSet: {Y,Z}+	PC Value Sets
T5CB2	S		Default value sets for PC data types

T5CB3	E	Parar: → T5CAA	PC parameter characteristics
T5CA4/5	E	Schema: {Y,Z}+	PC Schemas
T5CP6/7	E	Forms: {Y,Z}+	PC forms
T5CB8/9	E	RepTy: 9+	PC Report types
T5CBA	C		PC Schemas-Report types-Assignment
T5CBB	C		PC Output sequence-Report types-Assignment
T5CBC	C		PC Schemas-Output sequence-Assignment
T5CBD	G	Formu: {Y, Z}+	PC Form characteristics
T5CBE	G	OutTy: {Y, Z}+	Output Assignments
T5CBF	G	OutTy: {Y, Z}+	PC Individual processes
T5CBG/H	E	OutSq: {X,Y,Z}+	Output Sequences
T5CBI/J	E	GrpTy: {Y,Z}+	Group types
T5CBK/L	E	OutTy: {Y,Z}+	Output assignment categories
T5CBM	C		Output sequence steps
T5CBN	G	—	PC operation requirements
T5CBO	G	OutTy: {Y, Z}+	Additional selection parameters
T5CBP/Q	E	Calcn: {Y, Z}+	PC calculations
T5CBR/S	E	CView: {Y, Z}+	PC Online evaluation views
T5CBT	C		PC calculation control
T5CBU	C		Field selection for PC online evaluation view
T5CBW	C		Value Type Characteristics for Dialog
T5CBX	G	Schema: {Y,Z}+	PC schema settings
T5CBY/Z	S		PC contexts
T5CC0	E	Pplan: → T5CA4	Additional PC method steps
T5CC1	G	OutSq: → T5CBG	Output sequence characteristics

Explanation:

- + is a user-defined character (dependent on the respective domain)
- {...} is a character for this set
- ‚#‘ is a number (# = {0, ..., 9})

Example: {Y, Z}+ = Y0, Y1, ..., Y9, YA, YB, ..., YZ, Z0, ..., ZZ

Single/Multiple Step Tasks

You can select PF tasks using the task category TS/WS and identification code HRCHPK* (F4-Help).

The tasks are summarized in the task groups HRCHPK_BP, HRCHPK_LP and HRCHPK_ALL.

Domains for the Entity Copier

P02P0	Pension funds
P02V0	Parameter area determination groupings
P02V1	Processing schedules
P02V2	Parameter areas

Features

CHPK1	Determine EE grouping for method determination
CHPKB	Authorization group for PF account
CHPKP	EE grouping for field assignment infotype 0279
CHPKN	Number range for number range object HRCHPKONTO

Number Range Object

Use the number range object *HRCHPKONTO* to determine account numbers when creating individual PF accounts.

Area Menu

Use the area menu PACK to access PF management functions. It contains functions for payroll administrators and technicians, who maintain PF settings. You can use this as a template for individual area menus (maintain with transaction SE43N).

Address Group

In central address management, you have an address group PA01, for the pension fund.