Developing an Infotype in Personnel Administration

HELP.PAXX

Release 4.6C

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
💡	Note
🧭	Recommendation
SYN	Syntax

Contents

Developing an Infotype in Personnel Administration

This section provides you with an overview of infotype development for Personnel Administration. It covers the following issues:

- How are HR infotypes structured?
- How do you create your own infotypes?
- How do you modify infotypes included with the standard system?
- How do you supplement infotypes for fast data entry?
  - How do you adapt the interfaces within Personnel Administration?

In order to gain the most from this section, you are required to have prior knowledge of programming with *ABAP* and be acquainted with the *ABAP Screen Painter* and *Data Dictionary*.

Developing an Infotype in Personnel Administration

The Infotype Concept within Personnel Administration

Infotypes are used in *Personnel Administration* and *Recruitment.* An infotype is a group of object-based pieces of information on a particular area. The data stored in an infotype is always based on the personnel number of an employee or the applicant number of an applicant. In other words, an infotype record is always assigned to exactly one employee or applicant.

A four digit number **nnnn** is assigned to each infotype. This number uniquely identifies an infotype. The number range **9000** to **9999** is reserved for customer infotypes.

Transaction PM01 *Enhance Infotypes* enables you to create and edit infotypes.

See also:

Definition of an Infotype within the Data Dictionary [Page 8]

Module Pool for an Infotype [Page 14]

Screens for an Infotype [Page 16]

Dialog Module for an Infotype [Page 31]

Characteristics of an Infotype [Page 32]

Infotype Text Modules [Page 33]

Definition of an Infotype within the Data Dictionary

Each infotype **nnnn** requires at least two structures and one table:

- Structure **PSnnnn**

  Structure **PSnnnn** contains all of the infotype data fields.

- Transparent table **PAnnnn** and/or transparent table **PBnnnn**

  Transparent table **PAnnnn** is required if you want to use an infotype within Personnel Administration.

  If you want to use an infotype within Recruitment, transparent table **PBnnnn** is required.

- Structure **Pnnnn**

  Structure **Pnnnn** contains infotype key fields and all of the data fields from structure **PSnnnn**.

You might also need to define further structures or tables for specific infotypes.

You create the data definitions of these structures and tables manually in the Data Dictionary. Transaction PM01 *Dialogs in HR* does not support this step at this time.

In accordance with the distribution of infotype name ranges, objects **P9nnn**, **PS9nnn**, **PA9nnn** and **PB9nnn** are assigned to the customer name range.

> You want to develop an infotype with the number **9900** to be used within Personnel Administration only. The names of the structures and tables for this infotype are as follows:
>
> - PS9900(structure)
> - PA9900(transparent table)
> - P9900(structure)

See also:

Structure and Task of Structure PSnnnn [Page 9]

Structure and Task of Tables PAnnnn and PBnnnn [Page 10]

Structure and Task of Structure Pnnnn [Page 12]

Additional Structures for Screen Fields [Page 13]

Structure and Task of Data Field Structure PSnnnn

Each infotype includes data fields that are only stored within that particular infotype. The fields are required when the infotype's data structures and database tables are defined. The data fields are grouped together in structure **PSnnnn** to keep the definition as free of redundancy as possible. Structure **PSnnnn** can then be used as a substructure when further structures and tables are defined in the Data Dictionary.

Do not use the data type **INT4** for the field names of structure **PSnnnn**.

CI Include

The *Personnel Administration and Recruitment* infotypes contain a customer include **CI_nnnn** in the structure **PSnnnn**. You can include your customer fields in this include. This is then an enhancement of a standard SAP infotype [Page 43].

Using subtypes

If you want to divide an infotype into subtypes, you must assign structure **PSnnnn** a duplicate of key field **Pnnnn-SUBTY** in which the subtype is then stored. This field requires its own name and data element.

You must include the duplicate subtype field in the appropriate infotype screens. You can then make entries in this field.

You must also specify the name of the duplicate subtype field in the *Subtype field* field of table T582A when you maintain the infotype characteristics. Central infotype modules automatically write data to key field **Pnnnn-SUBTY** from the entries in this field. Key field **Pnnnn-SUBTY** does not appear on infotype screens.

A duplicate subtype field has the following advantages:

- Special check tables are used for the subtype assigned to the infotype.
- Field-specific documentation can be created for the subtype and then displayed using F1 help.

- Field **PS0006-ANSSA** for the address type (check table T591A)
- Field **PS0014-LGART** for the wage type (check table T512Z)

Structure and Task of Tables PAnnnn and PBnnnn

The data records of infotype **nnnn** are stored in database tables **PAnnnn** and **PBnnnn.** The area in which the infotype is used determines which of the tables you require.

- If you want to use your infotype within Personnel Administration, create table **PAnnnn**.
- If you want to use your infotype within Recruitment, create table **PBnnnn**.
- If you want to use your infotype within Personnel Administration and Recruitment, create both table **PAnnnn** and **PBnnnn**.

You must also specify the database tables you want to use in the infotype characteristics [Page 32] (table T777D).

The tables have the following structure:

Table PAnnnn

Field name	Key	Data element	Type	Length	Check table	Short text
MANDT	X	MANDT	CLNT	3	T000	Client
.INCLUDE	X	PAKEY				
.INCLUDE		PSHD1				
.INCLUDE		PS**nnnn**				

Table PBnnnn

Field name	Key	Data element	Type	Length	Check table	Short text
MANDT	X	MANDT	CLNT	3	T000	Client
.INCLUDE	X	PBKEY				
.INCLUDE		PSHD1				
.INCLUDE		PS**nnnn**				

The definition of the primary key is the only difference between the structure of the two tables. The primary key is determined on the basis of the *Client* field and substructures **PAKEY** and **PBKEY**.

Technical Settings of Database Tables

Database tables are read using the primary index. It is rarely necessary to create secondary indices.

When determining the logical memory parameters, enter the value **APPL0** in the *Data class* field. The value of the *size category* can vary, depending on how the infotype is used. For this reason, you must estimate the number of expected data records and then specify a suitable size category.

Tables **PAnnnn** and **PBnnnn** may not be buffered using the SAP database interface because the application programs must always work with current data. For this reason, you must select the *Buffering not allowed* checkbox in the *Buffering* frame.

Infotype data records are buffered within the HR applications, irrespective of the Data Dictionary settings.

You can enter changes to infotype records in the form of change documents using the infotype log creation function within HR. Report RPUAUD00 enables you to display these documents. It is rarely necessary to log data changes within the Data Dictionary.

Structure and Task of Structure Pnnnn

Structure `Pnnnn` contains the data fields of structure `PSnnnn` and the data fields included in all infotypes. It consists of substructures

- `PSHDR` and
- `PSnnnn`.

Structure `PSHDR` contains the substructures

- `PSKEY` and
- `PSHD1`.

Structure `Pnnnn` contains almost the same fields as tables `PAnnnn` and `PBnnnn`. There are differences in the included key fields (`PSKEY` instead of `PAKEY` and `PBKEY`). Furthermore, the *Client* field is not required in the structure.

Structure `Pnnnn` is used within reporting and the infotype module pools. It serves as an interface between the program and the database. If you require more information on using infotypes in reports, please refer to the Logical Database [Ext.] section within *Programming in HR*.

Customer infotypes are included automatically in logical database *PNP*.

Additional Structures for Screen Fields

When defining screen fields in the ABAP Screen Painter, do not specify structural data (such as the data type and length) directly when maintaining the screen. It is better to specify such data indirectly so that it is taken from the definition of objects in the Data Dictionary.

Individual fields are contained in various structures, depending on their meaning:

- Fields that are displayed for all infotypes are stored in structure **`RP50M`**.

  Such fields include, for example, the headers for single screens.

- Fields that are infotype-specific are stored in structure **`PSnnnn`**.

You might want a screen to include screen fields that are not yet included in a structure. If this is the case, you must create structure **`Znnnn`** in addition to structure **`PSnnnn`**. You can then use structure **`Znnnn`** to store all of the fields that must be displayed on the screen but are not yet included in a Data Dictionary structure. The name of the corresponding structure for infotypes in the standard system is **`Qnnnn`**.

⚠

Data Dictionary structures with the name **`Qnnnn`** are always stored within the SAP name range.

The employee’s form of address is stored in field **`P0002-ANRED`** with the value **`1`** for **`Mr`** and **`2`** for **`Ms`**. However, you want to be able to enter and display the terms **`Mr`** and **`Ms`** on the screen rather than their corresponding numerical values. If this is the case, you must use an additional field **`Q0002-ANREX`**.

Infotype Module Pool

A module pool should be used with each infotype. This module pool is the main program for the maintenance interface for the infotype.

The name of the program is **`MPnnnn00.`** Where **`P`** stands for Human Resources (personnel) and **`nnnn`** is the four-digit infotype number.

Infotype-specific includes

The main program only contains INCLUDE statements. If you create the main program using transaction PM01 *Dialogs in HR*, the system also creates the following four includes:

Name of include	The include contains
`MPnnnn10`	the PROGRAM statement and the declaration of common data objects
`MPnnnn20`	PBO modules for the screens
`MPnnnn30`	PAI modules for the screens
`MPnnnn40`	subroutines

⚠

All of the changes you make to module pool **`MPnnnn00`** or the includes listed here for an infotype within the standard system constitute modifications.

General includes

The system also inserts INCLUDE statements in the main program for the following includes:

Name of include	Use
`FP50PPSB`	Declaration of common data objects This data area is used as a buffer for imported infotype records and maintenance information. The variables specified in this area are used as export or import parameters when the infotype dialog module is accessed.
`MPPDAT00`	Declaration of common data objects
`MPPERS00`	Standard infotype modules
`MPPIRC00`	Definition of infotype return codes
`MPPREF00`	Definition of two data objects that contain the number of reference personnel numbers in structure P0031 or P0121

These includes contain standard functionality that must be offered by each infotype.

⚠

Do not change these includes! They are used by module pools for all infotypes.

Includes for infotypes with national-specific functions

Many infotypes require modules that apply to just one country. You must store them separately in their own includes for data definition, PBO and PAI modules, and subroutines. You then enter the *HR country indicator*, which is assigned in table T500L to the appropriate *country grouping*, at the end of the name of the corresponding include.

Infotype Screens

Each infotype has at least three screens:

- An initial screen
- A single screen
- A list screen

It is also possible to adjust screen control to replace the single or list screen with alternative screens. This enables you to use different single or list screens for an infotype. You might want to use additional single or list screens within HR, for example, to adapt an infotype screen to the requirements of a particular country.

See also:

Initial Screen [Page 17]

Single Screen [Page 18]

List Screen [Page 22]

Screen control

Screen control enables you to adapt the interface.

Infotype Screen Control [Page 25]

Fast data entry

Infotypes also require additional screens if you want to be able to maintain them using fast data entry.

Infotype Enhancement for Fast Data Entry [Page 50]

Initial Screen

The initial screen of an infotype is used as a technical interface between transactions within Human Resource Management and the infotype itself. It is accessed via the dialog module assigned to the infotype in question.

Screen `1000` is used as the initial screen for all infotypes. Screen `1000` of module pool **`MPMMMM00`** is used as a model.

The initial screen is processed in the background, i.e. the screen is not displayed even though it is processed.

The initial screen performs the following tasks:

- It performs general initialization procedures required by all infotypes.
- It accesses the single screen.
- It performs general processing steps once the single screen has been processed.

You must always create the initial screen using transaction PM01 *Enhance Infotypes.* The system then creates an initial screen with all of the functions required. Do not change the initial screen.

Single Screen

The single screen of an infotype is the interface between the system and the user. It enables you to

- create
- display, or
- maintain

individual infotype data records. Screen `2000` is usually used for the single screen. However, you can choose to use a different screen as a single screen. Screen `2000` of module pool `MPMMMM00` is used as a model for the single screen.

You can create your own single screens for infotypes included in the standard system. Customer-specific single screens are assigned to the name range `2900` to `2999`.

Entry checks

The values of the *Organizational Assignment* infotype (0001) that are valid at the beginning of the record's validity period, as well as the entries in tables T001, T001P, T500P and T503 that are valid in structure PSYST, enable you to carry out infotype-specific entry checks.

This means that the system does not need to read infotype 0001 or the two tables cited above. It is sufficient to include the tables in the TABLES statement.

Do not use the values in structures P0001 or P0002. These structures are used internally and are not always initialized.

Possible entries for screen fields

The system displays possible entries for all of the fields whose entry is checked against a table.

If you assign a check table that can be checked automatically to a field within the Data Dictionary, the system displays possible entries automatically.

Screen setup

The first six lines of the initial screen are the same for all infotypes:

Line number	Use
`1` to `3`	Header lines with data such as the personnel number
4	Empty line
5	FROM date, TO date, lock field, text field, the last person to make a change, and the date on which the last change was made
6	Empty line

Infotype-specific fields are included in lines 7 to 21.

The infotypes in the standard system then contain an include screen for customer enhancements.

All of the screen fields must be included in one frame. The screen field that contains the subtype assigned to the infotype, however, may appear above the first frame.

Tab Strips on the Single Screen

It is possible to include a tab strip control on the single screen. If you wish to do so, please note the following dependencies:

The flow logic of the tab strip control subscreen contains three modules. The modules are in the include MPPERS00. The modify_subscreen module must be included as the first module of the PBO.

The hidden_data_subscreen module must be the last module included in the PBO.

The input_status_subscreen must be included as the first module of the PAI. The module is called up in the same way as the input_status module on the single screen: all fields that may be maintained must be included in a CHAIN statement.

```
process before output.
  MODULE modify_subscreen.
* ... Subscreen specific PBO Module ...
  MODULE hidden_data_subscreen.

process after input.
  chain.
    field: ... all maintainable fields ...
      module input_status_subscreen on chain-request.
  endchain.
* ... Subscreen specific PAI Module ...
```

The function codes for the buttons on the tab strip control must be the same type as the function codes of a P button (local GUI function, function code is processed directly from GUI). No PAI will then be triggered by scrolling. If this is not possible, as validations are to take place when scrolling between the different screens, a new PBO must be triggered on the single screen before the last module, post_input_checks is executed.

Otherwise in the post_input_checks module, the function code field, fcode is deleted.

Flow Logic of Single Screen

If you create the single screen for the infotype using transaction PM01 *Enhance Infotypes*, the system prepares the flow logic.

The flow logic of infotypes within the standard system usually follows this pattern.

Action PBO

```
PROCESS BEFORE OUTPUT.


   MODULE BEFORE_OUTPUT.
   MODULE get_header_subscreen.
   CALL SUBSCREEN subscreen_header INCLUDING header_prog header_dynnr.
   MODULE Pnnnn.
   MODULE get_t582c_subscreen.
   CALL SUBSCREEN subscreen_t582c INCLUDING subscr_prog subscr_dynnr.
   MODULE HIDDEN_DATA.
```

You can carry out infotype-specific initialization procedures within PBO module `Pnnnn`. They enable you, for example, to fill the screen fields stored in structures `Qnnnn` and `Znnnn`.

If wage types are valuated indirectly, amount field `Q0014-BETRG` within the process logic of the *Recurring Payments and Deductions* infotype (0014) must be filled because the amount is not stored on the database.

You must not change PBO modules `BEFORE_OUTPUT` and `HIDDEN_DATA`.

Action PAI

```
PROCESS AFTER INPUT.


   MODULE EXIT AT EXIT-COMMAND.
   CHAIN.
      FIELD Pnnnn-feld1,...
      MODULE INPUT_STATUS ON CHAIN-REQUEST.
   ENDCHAIN.
```

PAI module `INPUT_STATUS` must be performed if the user makes an entry in a screen field. For this reason, all of the entry fields in the following chain must be counted. PAI module `INPUT_STATUS` sets internal system statuses: if a value is changed, for example, the infotype record must be stored later.

```
MODULE PRE_INPUT_CHECKS.
```

PAI module **PRE_INPUT_CHECKS** is used to process the function code before the entry check. If you choose the *Exit* function, for example, the system stops processing the current single screen.

Once module **PRE_INPUT_CHECKS** has been processed, you can carry out your own entry checks or call up your own PAI modules.

- You want an entry for the field **Pnnnn-feld1** to be validated against table Tnnn:

```
FIELD Pnnnn-feld1
SELECT * FROM TABLE Tnnn WHERE feld1 = Pnnnn-feld1
ON INPUT.
```

- You want to perform module **Modul_feld2** if the user makes an entry in field **Pnnnn-feld2**:

```
FIELD Pnnnn-feld2 ON INPUT MODULE Modul_feld2.
```

At this point, the entry checks must be complete. Once the following process has been carried out, it is no longer possible to change field contents.

PAI module **POST_INPUT_CHECKS** processes the function code after the entry checks. It also carries out general entry checks. The system checks, for example, whether the start date of the infotype record is before the end date of the record.

All fields on the screen in the following chain must be counted. All of the fields displayed on the single screen must be counted in the following chain.

```
CALL SUBSCREEN subscreen_t582c.
CHAIN.
   FIELD Pnnnn-feld1,RP50M-SPRPS,Tnnn-felda,... .
   MODULE POST_INPUT_CHECKS.
ENDCHAIN.
```

You must not change PAI modules **EXIT**, **INPUT_STATUS**, **PRE_INPUT_CHECKS** and **POST_INPUT_CHECKS**.

List Screen

The list screen of an infotype enables you to display all of a specific infotype's data records created for a personnel number.

Screen `3000` is usually used for the list screen. You can, however, choose to use a different screen as a list screen. Screen `3000` of module pool **`MPMMMM00`** is used as a model.

You can create your own list screens for infotypes included in the standard system. Customer-specific list screens are assigned to the name range `3900` to `3999`.

Screen setup

The list screen is divided into three areas:

- Lines 1 to 3 include the header lines.
- Lines 5 to 19 contain the list with the infotype records.

  Fields assigned to structure **`Pnnnn`** are usually displayed. If you want to display further information, such as long texts, in the list screen, you can maintain further fields for this purpose.

  Infotype records can be displayed in one or several lines within the list screen.
- Line 20 contains the *Choose* fields (**`RP50M-BEGDA, RP50M-ENDDA`**, **`RP50M-SUBTY,`** **`RP50M-ABGRD`** and **`RP50M-PAGEA`**).

  These fields enable you to select infotype records within the list in accordance with the validity period, subtype, delimitation date, or item in the list.

  With the exception of the *Delimitation date* field (**`RP50M-ABGRD`**), it should always be possible to make entries in these fields. The delimitation date in field **`RP50M-ABGRD`** should only be displayed on the list screen if the current function really is to delimit.

If you create the list screen using transaction PM01 *Enhance Infotypes*, the system sets the list screen up automatically. Fields assigned to structure **`Pnnnn`** are also included in the list screen.

If you do not use subtypes in the infotype, delete field **`RP50M-SUBTY.`**

Flow Logic of List Screen

If you create the single screen for the infotype using transaction PM01 *Enhance Infotypes*, the system prepares the flow logic.

The flow logic of infotypes within the standard system usually follows this pattern.

Action PBO

```
PROCESS BEFORE OUTPUT.
   MODULE BEFORE_OUTPUT.
   MODULE ASSIGN_TC3000.
   MODULE VARIATION_TC.
   LOOP.
      MODULE PSLIST.
      MODULE Pnnnn.
     ENDLOOP.
   MODULE GET_HEADER_SUBSCREEN.
...CALL SUBSCREEN SUBSCREEN_HEADER INCLUDING HEADER_PROG HEADER_DYNNR.
```

You can carry out infotype-specific initialization procedures within PBO module **Pnnnn**. This is the same module that is used for the single screen. If you require different infotype-specific initialization procedures for the list screen, you can determine that a different PBO module is accessed. This module must be called **PnnnnL**.

⚠

You must not change PBO module **PSLIST**.

Action PAI

```
PROCESS AFTER INPUT.
   MODULE EXIT AT EXIT-COMMAND.
   LOOP.
      FIELD RP50M-SELEC MODULE MARK ON REQUEST.
     ENDLOOP.
   CHAIN.
      FIELD RP50M-BEGDA.
      FIELD RP50M-ENDDA.
      FIELD RP50M-SUBTY.
      MODULE SELECT_FOR_LIST ON CHAIN-REQUEST.
   ENDCHAIN.
   FIELD RP50M-PAGEA ON REQUEST MODULE TOP_OF_LIST.
```

Flow Logic of List Screen

```
MODULE ADJUST_LIST_TC.

MODULE POST_INPUT_CHECKS.
```

⚠

You must not change PAI modules **`EXIT`**, **`MARK`**, **`SELECT_FOR_LIST`** and **`POST_INPUT_CHECKS`**.

Infotype Screen Control

When you create single screens and list screens using the ABAP Screen Painter, you determine the attributes of individual screens. However, the same screen is always used for various functions such as to create, display, maintain, and delete infotype records. For this reason, you cannot specify whether entries can be made in a screen field when maintaining the screen. It is also possible that particular screen fields must be hidden, depending on the employee's organizational data.

In other words, some attributes are not specified to be generally applicable; instead, they are specified during runtime. The appearance of the screens changes depending on the function chosen by the user or the data being processed.

- The same single screens are used for the *Display master data* and *Maintain master data* functions. However, it must only be possible to make entries in the fields if the *Maintain master data* function is chosen. For this reason, you must determine whether entries can be made in the fields or not depending on the function to be carried out.
- The screens used to enter company car data in the *Internal Control* infotype (0032) must be hidden for employees assigned to the employee group for pensioners.

Screen control functions have already been implemented in the main infotype program. These functions read the values from the modification groups for individual screen fields and set their attributes in accordance with the values. The meaning of the values for modification groups is determined in tables.

When you develop infotypes, you can effect screen control as follows:

- You can determine whether entries can be made in fields, or you can choose to hide screen fields, depending on the function to be carried out.

  The value in *Modification group 1* controls whether the screen fields are ready for input. You can also hide individual screen fields.

  It is also possible to hide individual screen fields.

  The value of *Modification group 1* must be maintained for all entry fields.

- You can use alternative screens, or determine whether entries can be made in fields, or hide screen fields using control data in table T588M.

  In connection with table T588M, the value of *Modification group 3* determines the activity and whether entries can be made in fields.

If you use both of the above to effect screen control for a screen field, screen control using table T588M has higher priority.

Modification group 2 is used internally.

Modification group 4 is not used in the standard system because it is reserved for customers. If you use this field, you are modifying the SAP standard system.

See also:

Screen Control Based on Function to be Performed [Page 27]

Infotype Screen Control

Screen Control Based on Control Data [Page 29]

Screen Control Based on Function to be Performed

If screen control is effected depending on the function to be performed, you can

- Determine whether entries can be made in individual screen fields, and
- Hide individual screen fields.

The ABAP Screen Painter enables you to maintain the value of *Modification group 1* for the screen fields in question. The value of *Modification group 1* must be maintained for all of the screen fields in which entries can be made.

The meaning of the values in *Modification group 1* is determined in table T589A. The following constants are defined in the standard system for determining whether entries can be made in screen fields:

Entry can be made in screen field for the function	**Hexadecimal constant for modification group 1**
Display	`001`
Change	`002`
Add and *Copy*	`004`
Delete	`008`
Lock/unlock	`010`

The following constants are defined in the standard system for hiding screen fields:

Screen field is hidden for the function	**Hexadecimal constant for modification group 1**
Delimit in list screen	`200`
Display in list screen and *Change* in list screen	`400`
Add and *Copy*	`800`

The value of *Modification group 1* is interpreted on a bit-by-bit basis. Several constants can be combined. This is done by adding the values. Note that you must maintain the value of *Modification group 1* in hexadecimal form.

- You want to be able to make entries in a screen field when the *Add* and *Change* functions are used. In this case, you must maintain value `006` in *Modification group 1*.
- You want to be able to make entries in a screen field for all functions. In this case, you must maintain value `00F` in *Modification group 1*.

Standard settings

In the case of certain screen fields for single or list screens, the setting that determines whether entries can be made or not is pre-specified for all infotypes. If you create the single or list screen using transaction PM01 *Dialogs in HR*, the system enters the correct value in *Modification group 1* for these screen fields.

  - Entries can usually be made in the fields **BEGDA** and **ENDDA** for all actions, apart from displaying records. For this reason, the attribute *Modification group 1* is assigned the value **00E** for these fields.

- *Modification group 1* has the value **800** for the fields **AEDTM** and **UNAME**. This ensures that these fields are hidden when a record is added.

- The list screen should allow entries to be made in the fields **RP50M-BEGDA**, **RP50M-ENDDA**, **RP50M-SUBTY** and **RP50M-PAGEA** so that records can be selected. These fields are assigned the value **00F** because it must be possible to make an entry for each operation.

  - The delimitation date in field **RP50M-ABGRD** should only be displayed on the list screen if the current function really is to delimit. For this reason, *Modification group 1* is maintained using value **400**.

  - It is only possible to select multiple records on the list screen if the display and delimit functions are used. Field **RP50M-SELEC**, which is contained in a loop, is assigned the value **009** for *Modification group 1*.

Screen Control Based on Control Data

If screen control is effected depending on control data, you can

- Replace the standard screen with an alternative screen
- Determine whether entries can be made in individual screen fields, or
- Hide individual screen fields.

Screen control can be effected

- Generally
- Depending on the employee's organizational data, or
- Depending on the subtype for the infotype record.

The ABAP Screen Painter enables you to maintain the value of *Modification group 3* for the screen fields in question

In *Modification group 3*, you assign a value between `001` and `050` to each screen field. Use the same value for screen fields that are modified in the same way. In the case of an input/output field, the same value is used as for the pertinent key word and a long text that may have been displayed. If screen fields cannot be modified using table T588M, assign the value **`SPACE`** in *Modification group 3*.

You then use table T588M to determine

- Whether and which alternative screens are used
- How the individual screen fields are modified.

If you require further information on screen control depending on control data, please refer to the IMG on *Personnel Administration* and read the documentation on the step *Determine screen modification* in the section entitled *Set Up User Interfaces*.

Infotype Interface Status

The interface for single and list screens is standard for all infotypes. A specific interface status is used depending on the function to be carried out. It is also possible for particular menu options or function keys defined in the interface status to be deactivated when certain functions are used.

The interface status is set in a PBO module included in the standard system. If you create your infotype using transaction PM01 *Enhance Infotypes*, the PBO module is accessed automatically by the flow logic of the infotype screens. For this reason, you do not need to program the interface status yourself.

The PBO module that sets the interface status can only function properly if the name and structure of the interface status to be used abide by SAP conventions. For this reason, you should also use transaction PM01 *Enhance Infotypes* to create the interface for your infotype.

List of required interface statuses:

Screen	**Interface status**	**Use of interface status for the function**
Single screen	DIS	*Display*
	MOD	*Change*
	DEL	*Delete*
	COP	*Copy*
	INS	*Add*
	EDQ	*Lock*
List screen	LIS0	*List screen/display*
	LIS1	*List screen/maintain*
	LIS9	*List screen/delimit*

The initial screen does not include an interface status.

Infotype Dialog Module

Each infotype requires a dialog module that represents the interface between the transactions used within Human Resource Management and the infotype itself. The name of the dialog module must be **`RP_nnnn`**, where nnnn stands for the number of the infotype.

The dialog module is assigned to an infotype when the dialog module is maintained. You must specify the name of the module pool and the number of the initial screen for the infotype.

The infotype is assigned to the dialog module in table T582A or by the name of the dialog module.

Infotype 0002 *Personal Data* uses module pool **`MP000200`** and screen `1000` as its initial screen. Therefore, this infotype requires a dialog module called **`RP_0002`** which accesses screen `1000` for module pool **`MP000200`**.

Infotype Characteristics

Infotype characteristics are determined by entries stored in various tables.

The following tables must be maintained for each infotype:

Name of table	Task
T582A	Basic infotype characteristics (database tables, single screen, list screen, time constraint, dialog module, and so on)
T582S	Infotype short texts
T777A	Technical Characteristics of Infotype (database table, dialog module, and so on)
T77ID	Name of data field structure (PSnnnn)

You can maintain tables T582A and T582S in view **V_T582A**. Tables T777D and T77ID are maintained automatically when the *Enhance Infotypes* transaction (PM01) is used.

You may also be required to maintain further tables:

Name of table	Task
T591A	Table T591A is used if the infotype is divided into subtypes. The subtype characteristics are determined in this table. You can use a different table instead of table T591A. The table used to store the subtype characteristics must be specified in table T582A as the subtype table.
T588G	Table T588G controls field-specific retroactive accounting.
T588M	Table T588M enables you to adapt infotype interfaces using screen control [Page 29]. (You can specify an alternative or subsequent screen; user-specific screen control is also possible).
T588B	Infotype menus
T588Z	Dynamic actions

The entries stored in these tables must be maintained manually. Transaction PM01 *Enhance Infotypes* enables you to maintain the basic infotype characteristics and to set up infotype menus.

Infotype Text Modules

The SAP System enables you to create a text module when entering master data for individual infotype data records. These text modules are stored in file *PCL1* under cluster ID *TX*.

To ensure that text modules can be created for an infotype, the *Text allowed* field (**`T582A-INFTX`**) must be flagged when the infotype characteristics are maintained (table T582A).

When you display or maintain an infotype record, you can display or maintain its text. To do this, in the individual screen for the infotype, choose *Edit → Display text* or *Edit → Maintain text.*

Displaying and maintaining text on the single screen

You can also display or maintain the first three lines of text on the infotype single screen.

If you want to use this functionality, simply adjust the single screen in question. You do not need to change the infotype structures or tables in the Data Dictionary.

The first three lines of text on the single screen for infotype 0019 *Monitoring of Dates* are displayed or can be maintained.

See also:

Setting Up a Single Screen for Displaying and Maintaining Text Modules [Page 34]

Setting Up a Single Screen for Displaying and Maintaining Text Modules

If you want to display or be able to maintain the first three lines on the single screen of your infotype, proceed as follows:

1. Change the display of the infotype single screen in question.

   Include the fields **RP50M-TEXT1**, **RP50M-TEXT2**, **RP50M-TEXT3** on the single screen.

   To ensure that entries can be made in these fields when the functions *Add* and *Change* are used, maintain the value **006** in *Modification group 1* for these three fields.

2. Enhance the flow logic for the action **PROCESS BEFORE OUTPUT**.

   Insert the module **GET_TEXT** behind the module **HIDDEN_DATA**. The module **GET_TEXT** is, therefore, accessed as the last module of this action.

3. Enhance the flow logic for the action **PROCESS AFTER INPUT**.

   Insert the following lines after the module **PRE_INPUT_CHECKS** and before the infotype-specific entry checks:

   ```
   CHAIN.
      FIELD: RP50M-TEXT1, RP50M-TEXT2, RP50M-TEXT3.
      MODULE UPDATE_TEXT ON CHAIN-REQUEST.
   ENDCHAIN.
   ```

   You must also include the fields **RP50M-TEXT1**, **RP50M-TEXT2**, **RP50M-TEXT3** in the chain for module **POST_INPUT_CHECKS** so that entries can be made in these fields when the message *W200 Please save your entry* is displayed.

4. Check that the *Text allowed* indicator has been set in table T582A.

   If this indicator is not set, the fields **RP50M-TEXT1**, **RP50M-TEXT2**, **RP50M-TEXT3** are hidden.

Result: You can maintain texts for your infotype. The first three lines of text are displayed on the single screen or can be maintained.

Guideline on Creating an Infotype

This section explains the steps which you must carry out to develop your own infotype. Please note that this documentation only covers the procedure for the implementation phase. Modeling and designing the infotype must be completed prior to implementation.

Development classes for customer developments

You must use customer-specific development classes for all of the sub-objects within your infotypes. It is important that you use customer-specific development classes and observe the naming conventions so that your developments are not lost when the system is upgraded.

If you use the *Enhance Infotypes* transaction (PM01), all objects must be in the same development classes.

Naming conventions

The number range `9000` to `9999` is reserved for customer infotypes.

Creating an infotype

You must first create all of the sub-objects that pertain to an infotype.

These sub-objects must offer particular standard functionality or have a particular structure. For this reason, you should use transaction PM01 *Enhance Infotypes* to create your sub-objects. This transaction supports you by creating infotype sub-objects with the required functionality and the correct structure. The transaction uses a copy template for this. The template consists of a module pool with the name `MPMMMM00` which contains the various includes, screens, and the CUA status.

SAP recommends that you start the *Enhance Infotypes* transaction (PM01).

When you have created your infotype, it is integrated in the *Personnel Administration* and *Recruitment* transactions. Your infotypes are also included automatically in logical database `PNP.`

Implementing infotype-specific functionality

Once you have created your infotype in the system, you can implement the infotype-specific functionality.

See also:

Creating an Infotype [Page 36]

Implementing Infotype-Specific Functions [Page 41]

Infotype Creation

New infotypes are created in four steps:

1. Create the infotype definition in the Data Dictionary.
2. Create a main program that contains standard infotype functionality.
3. Create the dialog module that accesses your infotype's initial screen.

Maintain the infotype characteristics. You can use the functions offered by the ABAP Dictionary to create the infotype definition in the Data Dictionary.

You can use transaction PM01 *Enhance Infotypes* to create the main program and the dialog module.

See also:

Creating an Infotype [Page 37]

Maintaining Infotype Characteristics [Page 40]

Creating an Infotype

If you want to create the main program for an infotype, proceed as follows:

1. Start transaction PM01, *Create Infotype* (transaction PM01).

   You access the *Create Infotype* screen.

2. In the *Infotype no.* field, enter the four-digit number of the infotype you want to create.

   When you specify the infotype number, please remember to enter any leading zeros.

3. In the *Subobjects* group box, flag *PS structure.*

4. Choose *Create*.

   The *Dictionary: Initial screen* appears*:*

   If you require further information on the structure and task of individual objects, please refer to the section Definition of an Infotype Within the Data Dictionary [Page 8].

   If you require information on how to use the Data Dictionary, please refer to *BC - ABAP Dictionary*.

5. Create the structure PSnnnn [Page 9].

6. Choose *Activate.*

7. Return to the *Create infotype* screen.

8. Choose *Create All*.

This creates the structure Pnnnn [Page 12] and the database tables for your infotype.

If you have flagged *Employee Infotype* for your infotype, table PAnnnn [Page 10] is created.

If you have flagged *Applicant Infotype* for your infotype, table PBnnnn [Page 10] is created.

If you want to use your infotype within Personnel Administration and Recruitment, both tables are created.

The following sub-objects are also created for your infotype:

- **Module pool**
  - **`MPnnnn00`** Module pool for infotype **`nnnn`**
  - **`MPnnnn10`** Include for module pool **`MPnnnn00`**
  - **`MPnnnn20`** Include for module pool **`MPnnnn00`**
  - **`MPnnnn30`** Include for module pool **`MPnnnn00`**
  - **`MPnnnn40`** Include for module pool **`MPnnnn00`**

  See also:

  Module Pool for an Infotype [Page 14]

- **Screens**
  - `MPnnnn00 1000` Initial screen for infotype **nnnn**
  - `MPnnnn00 2000` Single screen for infotype **nnnn**
  - `MPnnnn00 3000` List screen for infotype **nnnn**

  See also:

  Screens for an Infotype [Page 16]

- **Interfaces**

  The system creates an interface that contains all of the interface statuses required. A list of interface statuses is included in the section Interface Status for an Infotype [Page 30].

- **Dialog module RP_nnnn**
- **Entry in table T777D for the technical characteristics of an infotype.**
- **Entry in Table T77ID for the data field structure Psnnnn for the infotype.**

9. Create additional structures or tables if required.

Result: You have created an infotype.

Make the required settings to table T582A in Customizing for *Personnel Administration.*

See also:

Maintaining Infotype Characteristics [Page 40]

Creating infotypes with name range enhancement:

Note the following information when creating infotypes and proceed as follows:

1. If you are creating an infotype with a name range enhancement (/Company 1/9000, for example), make sure that your entries are overwritten by those of another imported infotype with name range enhancement (Partner 1/9000, for example), if the infotype number of the imported infotype is the same as your infotype. For this reason, make sure before you import infotypes with name range enhancements that there are no conflicts between the infotype numbers available and those that are to be imported.Start the Personnel Administration infotype copier (PM01)Enter the infotype number.
2. Choose *Utilities* → *Name range*.
3. Enter the name range reserved for your company in the *Name range* field.
4. To create further infotypes, follow the procedure described in step 3 above.

Maintaining Infotype Characteristics

When an infotype is created, the system does not automatically create the table entries that define the infotype characteristics (tables T582A and T582S). For this reason, you must create the appropriate entries in the tables yourself. View V_T582A enables you to maintain these tables.

If you want to maintain the characteristics of an infotype, proceed as follows:

1. Start transaction PM01, *Create Infotype,* by entering the transaction code.

   You access the *Create Infotype* screen*:*

2. In the *Infotype no.* field, enter the four-digit number of the infotype you want to create.

   When you specify the infotype number, please remember to enter any leading zeros.

3. Choose *IT characteristics*.

   This accesses the *Display View "Infotypes": Overview* screen.

4. Choose *Table view* → *Display* → *Change*.

   This accesses the *Change View "Infotypes": Overview* screen.

5. Create a new entry for your infotype.

   You can create new entries by

   – choosing *Edit* → *New entries* or

   – copying an infotype entry with similar characteristics.

     If you want to copy an existing entry, select the entry you want to copy and choose *Edit* → *Copy as*.

6. Check the entries in the individual fields.

Result: You have maintained the characteristics of your infotype.

Implementation of Infotype-Specific Functions

This step enables you to do the following, for example:

- Set up single and list screens individually
- Program your own initialization procedures for screen fields
- Program your own entry checks

You can use transaction PM01 *Enhance Infotypes* to edit the infotype. You can also use the *Object Browser* within the *ABAP Development Workbench*.

See also:

Editing an Infotype [Page 42]

Editing an Infotype

To edit an infotype using the *Enhance Infotypes* transaction (PM01), proceed as follows:

1. Start transaction PM01, *Enhance Infotype,* by entering the transaction code.
2. Enter the name of the interface format. Choose *Create.*

   This accesses the *Create Infotype* screen*:*
3. In the *Infotype no.* field, enter the four-digit number of the infotype you want to create.

   When you specify the infotype number, please remember to enter any leading zeros.
4. In the *Subobjects* group, flag the subobject that you want to edit.
5. Choose *Edit.*

Result: The system starts the *ABAP Editor, Screen Painter,* or *Menu Painter,* depending on which subobject was selected.

Enhancing an Infotype Included in the SAP Standard System

Use

The enhancement concept for infotypes within *Personnel Administration and Recruitment* offers you the following functions:

- You can perform additional customer validations.

  For more detailed information, please see *Extended Help* under the *Enhancement* transaction (transaction code CMOD).

- You can include additional fields in an infotype.

  The enhancement concept allows you to include additional fields in the data field structure Psnnn. You can then maintain these fields in the standard individual screens.

  See also:

  Enhancing Single Screens [Page 45]

  It is also possible to include additional fields in the standard list screens.

  See also:

  Enhancing List Screens [Page 46]

  If you incorporate additional fields in an infotype, these will be treated in the same way as the SAP standard fields in reporting, when creating the documents, and within dynamic events.

  The enhancement of an infotype in the SAP standard system does not cause problems during a release upgrade.

Constraints

- The following infotypes are excluded from the enhancement concept:
  - *Actions* infotype (0000)
  - *Additional Actions* (0302)
  - *Time Management* infotypes (2nnn)
  - *Applicant Actions* infotype (4000)
- The length of the data field structure `PSnnnn` and the CI include must not exceed 1500 bytes.
- If you include additional fields in the *Organizational Assignment* infotype, (0001), these will not appear on the logical database *PNP* as selection fields.

Enhancing an Infotype Included in the SAP Standard System

Delete enhancement for infotypes:

To delete the infotype enhancements, use the following procedure: reset enhancement:

1. Start the Personnel Administration infotype copier (PM01)
2. Enter the infotype number.
3. Choose *Enhance infotype.*
4. Choose *Enhancement* → *Delete enhancement.*
5. Start the Data Dictionary (SE11)
6. Enter the PS structure of the corresponding infotype in the *Data type* field (PSnnnn).
7. Choose *Display*.
8. Choose *Activate* and carry out the updates to the infotype database table that are necessary according to the activation log. The database table for Personnel Administration infotypes is PAnnnn, the database table for Recruitment infotypes is PBnnnn.

Enhancing a Single Screen

Procedure

1. Start transaction PM01, *Create Infotype,* by entering the transaction code.

   You access the *Create Infotype* screen*:*

2. Choose *Enhance Infotypes.*
3. In the *Infotype no.* field, enter the four-digit number of the infotype you want to create.

   When you specify the infotype number, please remember to enter any leading zeros.

   ⚠

   It is not possible to enhance a single screen for the *Actions* infotype, 0000, or the *Time Management* infotypes.

4. In the *Sub-Objects* group box, flag *CI Include.*
5. Choose *Create*.

   The *Dictionary: Initial Screen* is displayed.

6. Create the CI include.
7. Choose *Activate.*
8. Return to the *Create Infotype* screen.
9. Choose *Create All*.

Result

You have included additional fields in the standard single field for an infotype.

Enhancing a List Screen

It is not intended that you make enhancements to the list screen in the Standard system. If you make an enhancement to the list screen, this represents a modification.

Procedure

1. Start transaction PM01, *Create Infotype,* by entering the transaction code.

   You access the *Create Infotype* screen.

2. Choose *List Screen.*
3. In the *Infotype no.* field, enter the four-digit number of the infotype you want to create.

   When you specify the infotype number, please remember to enter any leading zeros.

4. In the *Screen Number* field, enter the screen number of the list screen you want to enhance.
5. Choose *Create*.

   The *Dictionary: Initial screen* appears*:*

6. Create the list screen structure.
7. Choose *Activate.*
8. Return to the *Enhance List Screen* in the *Enhance Infotypes* transaction (PM01).
9. Choose *Create All*.

Result

You have included additional fields in the standard list screen for an infotype.

The additional fields are displayed on the list screen, however, they contain no data.

The fields can be filled in the FORM routine FILL-LISTSTRUCT in the generated program ZPnnnn00. The FORM routine is called for each data record in the list.

Structure ZPLIS is identified when it is generated with a TABLES statement in the program ZPnnnn00.

The fields can be filled from the Pnnnn structure or by reading text tables.

Deleting the additional fields

To delete the additional fields in the standard list screen, display the *Enhance List Screen* from the *Enhance Infotypes* transaction, then choose *List screen → Delete enhancement.*

The fields in structure ZPLISnnnn are removed from the standard list screen.

Modifying an Infotype Included in the Standard System

This section contains information on how to modify infotypes included in the standard system. Note that the following information is only partially applicable to *Time Management* infotypes.

As of Release 4.0A, the modifications described here are rarely used. The required functions are made available by the enhancement concept for infotypes in *Personnel Administration* and *Recruitment.*

For more information on the enhancement concept, see Enhancement of a Standard SAP Infotype [Page 43].

For this reason, you must check and possibly adapt your modifications after each release upgrade.

Modification options

You can add new sub-objects to infotypes included in the standard system. You can create

- customer-specific includes and
- customer-specific screens for an existing infotype.

You can use screen control [Page 29] together with table T588M to replace the infotype screens with your own.

If you want to add customer-specific includes to an infotype, you must change module pool **`MPnnnn00`** for the infotype.

> The structure or functions for the standard SAP infotypes can change in a Release upgrade.
>
> The customer modifications described in this section are not compatible with such changes. For this reason, you must check and, if necessary, modify, your modifications after each release upgrade.

Development classes for modifications

If you add new sub-objects to an infotype, you must always use customer-specific development classes. It is important that you use customer-specific development classes and observe the naming conventions so that your developments are not lost when the system is upgraded.

Naming conventions

Customer-specific sub-objects are assigned to the following name ranges:

Subobject	**Name range**
Customer-specific single screens	`2900` to `2999`
Customer-specific list screens	`3900` to `3999`
Includes for customer-specific data declarations	**`MPnnnn5x`**
Includes for customer-specific PBO modules	**`MPnnnn6x`**
Includes for customer-specific PAI modules	**`MPnnnn7x`**

Includes for customer-specific subroutines	**`MPnnnn8x`**

Within the names of the includes, **`nnnn`** represents the number of the infotype you want to modify. The last character **`x`** can be defined as you wish.

Infotype Enhancement for Fast Data Entry

If you want to use fast entry functions for your infotype within Personnel Administration, you must enhance it. Different types of enhancement can be effected, depending on the type of fast data entry you require.

You can adapt your infotype for the following types of fast data entry:

- Fast entry of master data (transaction code PA70)

  Infotype Enhancement for Fast Entry of Master Data [Page 51]

- Fast entry of action data (transaction code PA42)

  Infotype Enhancement for Fast Entry of Action Data [Page 54]

Infotype Enhancement for Fast Entry of Master Data

The fast entry of master data function enables you to create and maintain the same infotype and/or subtype record for more than one employee at the same time. If you require further information on the fast entry of master data, please refer to the *Personnel Procedures - Master Data* section within *PA Personnel Administration*. Please see,

See also:

Preparing an Infotype for Fast Entry of Master Data [Page 52]

Preparing an Infotype for Fast Entry of Master Data

If you want to be able to enter data fast for more than one person in infotype **nnnn**, carry out the following steps:

1. Use transaction PM01 *Enhance Infotypes* to copy screens **4000** and **5000** of module pool **MPMMMM00** to module pool **Mpnnnn00**.

    Screen **5000** is used for the fast entry of infotype records. Screen **4000** is required for entering default values if you want to use the *Create with default* function for fast data entry.

2. Edit screens **4000** and **5000** for module pool **MPnnnn00** in the *ABAP Screen Painter* (transaction code SE51).

    Include the fields in the screens that must be suggested as a default, or maintained, for fast data entry.

    Enter the value **006** in *modification group 1* of the entry fields.

    ⚠

    You must delete all of the indicators for automatic foreign key checking in screen **5000**.

    You can include entry checks for default values in the flow logic of screen **4000**, if required.

3. Check whether you have already copied the interface for both screens.

    The following interface statuses are required for the fast entry of master data:

    – **FINS** *Create*

    – **FMOD** *Change*

    – **FEDQ** *Lock/unlock*

    – **FDEL** *Delete*

4. Include the include **MPPERSFS** in module pool **MPnnnn00**.

5. Use transaction PM01 to copy include **MPMMMMFS** to include **MPnnnnFS** and then include the include **MPnnnnFS** in module pool **MPnnnn00**.

    Add the fields that are displayed as default values to the positions at which they are required.

    – For the definition of field string PROPOSED_VALUES

    – In subroutine DNNNN.

6. Maintain table T588R (*Selection Reports for Fast Data Entry*) and T588Q (*Screen Types for Fast Data Entry*) for the infotype.

    For further information on how to maintain these tables, see the IMG for *Personnel Administration* under *Setting Up Procedures* → *Infotype Menus* → *Set up infotype menu for fast data entry*.

7. Include the infotype in the menu for transaction PA70.

For further information on how to maintain these tables, see the IMG for *Personnel Administration* under *Setting Up Procedures → Infotype Menus → Change infotype menu*.

The following infotype menus are fixed for fast data entry:

- `07` fast entry of master data, and
- `09` fast entry of time data.

Infotype Enhancement for Fast Entry of Action Data

Transaction PA42

You can define fast data entry for each action in table T529A.

You must enter the required fields on the fast entry screen for each infotype assigned to the info group of the action. The majority of infotypes are already assigned default values from tables.They are not usually specified on the screen.

The specified data is written to internal table PROPOSED_VALUES, which is passed on to infotype 0000 *Actions*.

The action infotype accesses the infotypes assigned to the info group one after the other, and distributes the data entered on the fast entry screen.

Infotype dialogs are usually processed in the background.Individual infotype screens are only displayed if errors occur.

You can, however, use the fast entry screen to determine whether an infotype is processed in the foreground if, for example, you need to enter additional data for your employee, or data that deviates from the standard, that cannot be entered on the fast entry screen.

Definition of Entry Screen

Copy template

The following are used as copy templates:

- Program SAPMP5X0 screen 0100
  for national versions of the hiring action; 'X' must be replaced accordingly.
- Program SAPMP50F screen 0100
  for actions that do not maintain national infotypes; further screens can be added.

Field selection

The fast entry screen for actions must be compatible with the info group that pertains to the action.

If additional fields are included in the fast entry screen, they cannot be passed on to the infotypes and stored.

If fields are missing for infotypes processed by the action, the infotype single screen is displayed with the appropriate error messages.

Three different types of field are maintained on the entry screen:

- Fields from Pnnnn infotype structures (P fields)
  The contents of these fields are passed on to the appropriate infotypes where they are validated and stored.
  These fields are assigned the value '001' in modif group 1.
- Screen fields from Qnnnn structures (Q fields)
  These fields are used like the infotype single screens when the external format is different from the internal format.
  Each Q field must have an invisible P field on the screen which contains the field contents in the internal format, since only the contents of P fields can be passed on to the infotypes.

  Field Q0002-ANREX ('Mr', 'Ms') has invisible field P0002-ANRED ('1', '2') with the value '001' in modif group 1.
  Q fields are not assigned a value in modif group 1.

- Checkboxes for processing single infotype screens if additional data exists, abbreviated as 'additional fields' from now on.
  These fields are assigned the value '002' in modif group 1.

Process Logic

Module before output (PROCESS BEFORE OUTPUT)

MODULE INIT OUTPUT

The F status 'FAST' is set in the module INIT.
If integration to RP_PLAN is active, particular fields (P0001-KOSTL, P0001-ORGEH, P0001-STELL) are not changed to input fields.

Module after input (PROCESS AFTER INPUT)

MODULE EXIT

You leave the dialog when you enter ‘E’ in the OK code.

MODULE PERNR

The assigned personnel number is locked, the data is read.

MODULE FCODE

Function code processing:
With the ‘Save’ function code, the data is placed in the internal tables PROPOSED_VALUES and the infotype 0000 is called.

The system checks whether the specified personnel number exists. If this is the case, the data is not saved.

Generating Coding

Transaction PM01 enables you to generate coding from the entry screen.

- Data definition

  Infotype structures Pnnnn and Qnnnn and additional fields ADD_nnnn are declared in include member MPxxxTOP.

- Output modules

  The MODULE and ENDMODULE statements of the output modules are included in include member MPxxxO00.

- Input modules

  The MODULE and ENDMODULE statements of the input modules are included in include member MPxxxI00.

- Subroutines

  Routine FORM FILL_PROPOSED_VALUES_nnnn (nnnn = screen number) is then included in include member MPxxxF00.

  Within this routine, internal table PROPOSED_VALUES is filled with the names and contents of fields processed by the action.
  Modif group 1 of the fields is interpreted on the entry screen.

Procedure

A new entry screen is created for a module pool.

The includes with data definitions, output and input modules, and subroutines are supplemented accordingly.

An entry screen is changed for which coding already exists.

The includes with data definitions, output modules, and input modules are supplemented with new objects.

If fields or modules are deleted, the corresponding objects are not deleted from the includes because they might be used by other screens.

If routine FILL_PROPOSED_VALUES_nnnn already exists for current screen nnnn, a dialog box is displayed with a confirmation prompt and the routine is rebuilt.

Table Maintenance

In table T588A, an action menu, which is maintained in view V_588B_M, is assigned to the fast entry transaction PA42. Only actions with fast entry functions are permitted.

Within table T529F, the name of a dialog module in which the dialog module is determined is assigned to the actions with fast data entry.

Within table TDCT, an entry is included for the dialog module.

Adapting Interfaces within Personnel Administration

You can adapt interfaces and procedures within Personnel Administration as follows:

- You can determine individual headers for single and list infotype screens.

  For further information, see the IMG for *Personnel Administration* under *Adapting Interfaces within Personnel Administration → Define Screen Header*:

- You can determine alternative single and list screens, hide individual screen fields, and specify whether entries can be made in fields or not, by using table T588M to effect screen control with the aid of control data.

  For further information, see Screen Control [Page 29].

- You can include an infotype in an infotype menu for maintaining and displaying HR master data.

  For further information on how to maintain these tables, see the IMG for *Personnel Administration* under *Setting Up Procedures → Infotype Menus → Change infotype menu*.

- You can include an infotype in a personnel action.

  For further information, see the IMG for *Personnel Administration* under *Setting Up Procedures → Actions*.