If you are an SAP programmer working with more than just the SAP Human Resources (HR) module, you know that, by comparison, HR is one of the most demanding and challenging of SAP’s functional components. SAP HR has its own idiosyncrasies that you must understand in order to develop effective HR solutions, and experienced programmers will have an easier job if they have a thorough understanding of the tools and HR rules available to aid in HR development. In fact, there is not much information available at all about SAP HR development — I hope this article will at least partly fill this gap.

If you are an experienced programmer but have little or no HR knowledge, you can develop HR solutions, but the results of your work often may be difficult to manage. For example, personnel administration reports can be prepared in ABAP without using logical database PNP, a common enough practice. But the PNP logical database supports standard HR selection screens, simultaneous infotype processing (logical data views), and authorizations — everything that the programmer would otherwise have to prepare manually. Similarly, infotype modifications made without using HR tools such as transaction PM01 (i.e., using program repairs instead of standard enhancements) can complicate situations and cause additional work and difficulty during upgrades, since manual changes may be required for the modified programs.

The same programming challenges tend to arise in most SAP HR implementations. There are a few common questions, which when answered, can pave the way to a successful HR implementation. In this article you will find the answers to these common questions:

- What are infotypes?
- How can I enhance standard infotypes?
- What are HR “features” and how can I use them in enhancements?
• How can I use HR function module exits to influence infotype processing logic?
• How do I develop a custom infotype when enhancing a standard infotype isn’t enough to meet my needs?
• What are “dynamic actions” and how can I use them in infotype maintenance?

**Note!**

To take full advantage of the information in this article, it would be helpful to have more than just a basic knowledge of ABAP programming, the ABAP Dictionary, and the ABAP Workbench Screen Painter.

This article starts by exploring infotype functionality. Next, you will find out more about personnel actions, which are a useful tool for HR specialists, to help you better understand the SAP HR world and its programming requirements. Since most HR implementation teams need to enhance one or more of the standard SAP-provided infotypes (e.g., to add fields or to change the PBO or PAI for the infotype program), or to create new infotypes (e.g., when no standard infotype satisfies a particular requirement), I’ll then cover how to perform each of these tasks using the Personnel Administration (PA) infotype as an example. I will also show you how to use “dynamic actions” to help simplify infotype data maintenance. At the end of this article, you will hopefully have a solid foundational knowledge of HR programming that will enable you to mix ABAP coding and HR configuration in the most effective way.

**SAP HR — a user’s and a programmer’s perspective**

To understand advanced HR programming aspects, you need to have a basic knowledge of HR functionality. This section presents a short review of the HR infotype world from both the user’s and programmer’s perspective.

Experienced programmers will also find interesting aspects of the infotype “time constraints” concept. If you are already an HR expert, feel free to skip ahead.

**Infotypes**

The SAP HR module consists of several components (submodules). The Personnel Administration (PA), Organizational Management (OM), Personnel Time Management (PT), and Payroll (PY) components play a role in most HR implementations; of these, the most commonly used is PA, which I focus on in this article.

The central entity of an integrated HR system is the “employee” (or “candidate,” in a Recruitment context). In the SAP system, employees are identified by an eight-digit personnel number. These numbers follow the individual throughout his or her entire employment history (and even after termination, so there is no possibility of reusing a personnel number).

Personnel data is stored in “infotypes.” From the user’s point of view, an infotype is a group of logically connected data (e.g., Personal Data, Organizational Assignment, Addresses). From the programmer’s point of view, however, an infotype is a complex concept that aggregates transparent tables, structures, module pools, and technical characteristics into several tables.

There is a distinction between different kinds of infotypes — for example, PA infotypes describe the employee, while OM infotypes describe organizational objects, such as organizational unit, position, or job. This distinction is important not just from a user’s point of view; there are some technical nuances in the Data Dictionary representation for various infotypes that make them different, and that programmers need to understand when creating or enhancing infotypes. The download available at www.SAPpro.com takes a look at how creating and enhancing OM infotypes, for example, differs from creating and enhancing PA infotypes.

Relationships also exist between various infotypes. Figure 1 and Figure 2, for example, show the relationship between the PA and OM infotypes — the

---

1 All screenshots in this article were generated using data from the SAP IDES training system.
Figure 1  Sample PA infotype record describing employee 1331

Figure 2  Sample OM infotype record describing the PC service technician position
planned compensation for the position of the employee assigned in the Organizational Assignment (PA) infotype is defined in the Planned Compensation (OM) infotype.

Each infotype is identified by a four-digit number — for example, 0000 (Actions), 0001 (Organizational Assignment), 0002 (Personal Data), etc. The number range 9000–9999 is reserved for customer-defined infotypes. Some infotypes have subtypes that further categorize information — for example, infotype 0006 (Addresses) may have many subtypes, including Permanent Residence, Temporary Residence, Emergency Address, and so on. Subtypes can be configured (you can define new subtypes and change their names).

Customers generally use only a small number of the available standard infotypes. Although the standard SAP system contains many infotypes, there is often a requirement that can be satisfied only by defining a new infotype — mainly in PA, which is the example I use in the article to illustrate this task. Let’s take a closer look at PA infotypes.

### PA infotypes — what you should know

PA infotypes have important “time” attributes:

- Each infotype has a “validity period” defined by a beginning date and an ending date.
- Relationships between infotype records are defined by “time constraints” (see Figure 3). Time constraints in PA define the relationships between data records of the same infotype (the possible existence of a few records of a given infotype for the same employee in the same time period).

The implementation of time constraints ensures personnel data integrity. For example, the employee should have only one valid version of personal data at any given time. Other examples of the application of time constraints include the following:

- The employee must have only one organizational assignment at any one time during his or her activity in the organization; therefore infotype 0001 (Organizational Assignment) has time constraint 1.

### Table: Time constraint indicators for PA infotypes

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One valid infotype record must exist while the employee works at the organization. Creating a new record delimits the former record so that there is no gap or overlap. Examples of such infotypes are 0001 (Organizational Assignment) and 0002 (Personal Data) — these infotype records must exist at all times.</td>
</tr>
<tr>
<td>2</td>
<td>Similar to time constraint 1, overlapping is not allowed. In this case, however, the existence of infotype records at all times is not mandatory — in other words, only one valid record may exist at any one time, but there can be either one or no records. Example infotypes are 0045 (Loans) and 0017 (Travel Privileges) — the employee may or may not receive the company loan; the employee may or may not have travel privileges.</td>
</tr>
<tr>
<td>3</td>
<td>More than one record may exist concurrently. Gaps are allowed. Examples include infotype 0022 (Education) and 0023 (Other/Previous Employers).</td>
</tr>
<tr>
<td>A</td>
<td>Only one infotype record exists from 01/01/1800 through 12/31/9999 for each employee. The system creates the record automatically for an employee at hiring or when applicant data is entered, and it cannot be deleted. Examples include 0003 (Payroll Status) and 4004 (Applicant Activity Status).</td>
</tr>
<tr>
<td>B</td>
<td>No more than one record may exist for this infotype. It is valid from 01/01/1800 to 12/31/9999, and it may not be split (such that more than one record exists at any given time). Records may be deleted, however. Example infotypes are 0031 (Reference Personnel Numbers) and 0130 (Test Procedures).</td>
</tr>
<tr>
<td>T</td>
<td>Time constraint varies depending on the subtype. For example, infotype 0006 (Addresses) has time constraint 1 for subtype 1 (Permanent Residence) or time constraint 3 for subtype 2 (Temporary Residence). In this case, time constraints are assigned to the subtypes in view V_T591A.</td>
</tr>
</tbody>
</table>
- The employee may, but need not have, a single company loan; therefore infotype 0045 (Loans) has time constraint 2
- The employee may have many skills at the same time; therefore infotype 0024 (Skills) has time constraint 3.

**PA personnel actions**

To help simplify the maintenance of employee data, the PA component provides configurable groupings of related infotypes for common procedures. These groupings are called “personnel actions.” The most frequently used personnel actions are Hiring, Organizational Reassignment, and Termination. For example, the Hiring action may (depending on the configuration) group the following infotypes: 0002 (Personal Data), 0001 (Organizational Assignment), 0006 (Addresses), 0007 (Planned Working Time), 0008 (Basic Pay), and so on. When a personnel action is invoked, the user is led through the different screens of the infotypes assigned to this action.

There are some challenges involved in using personnel actions:

- Propagating default values from the current infotype screen to the next infotype screen
- Creating a new infotype data record (create a new entry for a given employee) in the background during the personnel action
- Invoking a procedure during the action according to data input by the user

Of course, there are several answers to these challenges, but one solution minimizes the risk — “dynamic actions.” In this article, I will show you how to use this solution.

**Understanding infotypes**

Before you can effectively use the tools that support the creation and modification of PA infotypes, you need to understand the infotype model and how it works. Here, I take you through the elements that constitute infotypes, and then I briefly introduce the basic transaction used to maintain them.

**The infotype architecture**

Each infotype is implemented using the definitions of several structures and tables in the Data Dictionary. There are two structures created for each infotype:

- PSnnnn — This structure defines the data fields specific to a given infotype (nnnn stands for the infotype number). The length of structure PSnnnn cannot exceed 2,000 bytes (in Release 4.7).
- Pnnnn — This structure contains key and control fields, as well as the structure of PSnnnn. Pnnnn serves as an interface for dialog programs and reports that use infotype data.

Since all PA infotypes should have a similar interface, the key and control fields of structure Pnnnn are defined in structure PSHDR as two structures: PSKEY and PSHD1. The meaning of each component of PSKEY and PSHD1 is explained in Figure 4. While data field definitions in the PSKEY structure are often used by programmers, the use of PSHD1 is limited mainly to reports showing “last change” attributes of the infotype (e.g., date of last change, last user who changed the record, etc.).

<table>
<thead>
<tr>
<th>Structure</th>
<th>Data element</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSKEY</td>
<td>PERNR</td>
<td>Personnel Number; uniquely identifies the person</td>
</tr>
<tr>
<td></td>
<td>INFTY</td>
<td>Infotype Number (four digits)</td>
</tr>
<tr>
<td></td>
<td>SUBTY</td>
<td>Subtype</td>
</tr>
</tbody>
</table>

Figure 4 PSHDR structures PSKEY and PSHD1 with key fields

Continues on next page
Data for each PA infotype is held in transparent table PAnnnn. This table has a structure similar to Pnnnn with two main differences: table PAnnnn contains the MANDT (client) field (personnel data should be client-dependent), and table PAnnnn does not contain the INFTY (infotype number) component among its key fields.

Let’s look at PA infotype 0006 (Addresses). Most employees have at least one record of this infotype (a permanent or temporary residence, for example). It is also possible that employees may have multiple versions of such records (e.g., permanent residence, mailing address, emergency address, etc.). Technically speaking, there are several subtypes of infotype 0006 representing different types of addresses (the list of possible subtypes can be configured). The single screen\(^2\) for infotype 0006 may look like Figure 5 (bearing in mind that there can be differences for some country-specific versions of HR).

The infotype table PA0006 is defined by the MANDT field and three structures (see Figure 6). The structures PAKEY and PSHD1 are the same

---

\(^2\) The single screen is the interface presented to the user for maintaining and displaying infotype records. In addition to a single screen, an infotype also has an initial screen, which initializes the infotype and calls the single screen, and a list screen, which displays all records associated with an employee number.
**Figure 5**  Single screen for infotype 0006

**Figure 6**  Table PA0006 representing infotype 0006 displayed in the Data Dictionary
for all infotypes; structure PS0006 includes only specific fields for this infotype (see Figure 7).

Structures of type PSnnnn usually contain a customer include, CI_nnnn, which can be used to enhance infotypes with customer-defined fields. I’ll discuss the use of such fields later in this article.

Maintaining infotypes

The basic transaction used to maintain infotypes is PA30 (Maintain HR Master Data).\(^3\) Let’s look at an example to help you understand how this transaction works. Let’s say we want to maintain address data for the sample employee number 10966 (Andrew Anderson). Suppose we want to check the current permanent residence data for this employee. Simply run transaction PA30 and enter the employee number in the Personnel no. field at the top of the screen. Then on the Core Employee Info. tab, select infotype number 0006 (Addresses) by selecting Addresses from the Infotype text list, choose Today in the Period frame, and enter subtype 1 (Permanent residence) in the STy field in the Direct selection frame (see Figure 8).

---

\(^3\) Transaction PA20 (Display HR Master Data) has a similar interface but is used only for displaying data.
Clicking on the Display button ( ) or pressing F7 takes you to the single screen for infotype 0006 (shown in Figure 5).

**Note!**

Standard screen 2000 is usually used for the single screen for infotype 0006, but that’s not a rule. It’s possible to use a customer-specific screen (numbered in the range 2900–2999). Use table T588M to assign your custom screen to module pool MPnnnn00 (where nnnn is the infotype number).

As stated before, structure Pnnnn is used as an interface between the program and the database. This interface is managed by program MPnnnn00. In the example, the single screen shown in Figure 5 is managed by program MP000600. The screen shows fields through the structure P0006. For some infotypes it may be necessary to show fields not included in structure Pnnnn. For example, the single screen for infotype 0001 has a field Q0001-KOSTX (cost center text). Structure Qnnnn is used in such cases. When working with customer infotypes, structure Znnnn should be defined for fields not included in structure Pnnnn.

![Figure 8](image-url) Transaction PA30 for HR master data maintenance
Tip!

The standard SAP system provides many country-specific infotypes. To make life easier for users, you can limit the number of infotypes visible in transactions PA20 or PA30 by configuring table T582L. (For additional information, in the IMG navigate to Personnel Management → Personnel Administration → Customizing Procedures → Assign infotypes to countries.)

Recall that all infotypes have validity dates — you can see them near the top of Figure 5 (the “Start” and “to” fields). Notice that the “to” (end) date has been set to 12/31/9999 (HR consultants sometimes call this date “infinity”), which means that this is the latest address for this employee (there are no other records for permanent residence valid subsequent to 06/01/1999).

Suppose the example employee changed his permanent residence address on 10/01/2005 from 998 Buster Lane to 990 Buster Lane. To input this information in a way that preserves a history of prior addresses, run transaction PA30, enter the personnel number, infotype, and subtype as before, and click on the Create button ( ) or press F5. After entering and saving the new address data, you will see the message “Record valid from 06/01/1999 to 12/31/9999 delimited at end,” which means that the validity date of the former record now has been changed to end on 09/30/2005.

Now, suppose we want to see all the address records for this employee. From the PA30 screen (Figure 8), click on the Overview button ( ) to display the list screen shown in Figure 9. If you check the record with the start date 06/01/1999, you will see that the end date has been set to 09/30/2005 (the day before the start date of the new record, 10/01/2005).

Note!

Standard screen 3000 is usually used for list screens; customer-defined list screens have the number range 3900–3999.
Note!

Each infotype has an initial screen that is processed in the background. The initial screen initializes infotype dialog processing and accesses the single screen. Its ID is always 1000. You should not change this screen.

Now that you understand the underlying structure of infotypes and are familiar with the basic transaction for maintaining them, let’s take a look at how to enhance standard SAP-provided infotypes to meet your specific needs.

Enhancing standard infotypes

Let’s say that we need to enhance a standard infotype with additional customer fields. For example, suppose we want to add three new fields in PA infotype 0006:

- Travel expenses (travel from place of living to work)
- Currency
- A checkbox indicating whether this amount should be paid in regular payroll

All this can be done using standard SAP transaction PM01 (Enhance Standard Infotypes), without any manual insertion of ABAP code:

1. Start transaction PM01, choose Enhance, and enter the number of the infotype to be enhanced (0006), as shown in Figure 10.

2. Select CI include in the Subobjects frame and click on Create, which takes you to the screen for structure maintenance (Figure 11 on the next page).

3. Create the CI include structure CI_P0006 with the three additional fields shown in Figure 11, and then activate it.

Figure 10  PM01 screen for standard infotype enhancement
Note!

• Double-clicking on component type ZFLAG shows it has a standard domain XFELD.

• Component ZZAMT is assigned reference field ZZWAE on the Currency/quantity fields tab.

• Check table TCURC is assigned to component ZZWAE on the Entry help/check tab.

4. Return to the PM01 screen and click on Create All ( ).

5. To display the ZZPAY field on screen 0200 as a checkbox, go to the ABAP Workbench Screen Painter, choose screen 0200 (ZP000600), select field P0006-ZZPAY, and simply follow the menu path Edit → Convert → Checkbox.

Now run transaction PA30 and invoke the single screen for infotype 0006. As you see in Figure 12, the system has added new fields at the bottom of the screen in the Additional fields section. In particular, a new module pool ZP000600 with several includes (ZP000610, ZP000620, ZP000630, ZP000640) has been generated, along with screen 0200 (ZP000600), which is embedded as a sub screen of a single screen of infotype 0006 (MP000600). The sub screen (the screen number and the module pool) to be included in

Figure 11  Sample CI_P0006 structure
the single screen is assigned to the module pool MPnmmn00 in table T582C. All this happens automatically and is done by transaction PM01!

As it stands, the newly added fields are shown for all subtypes of infotype 0006. Let’s say that we want to maintain these fields only for subtype 1 (Permanent Residence). How can we do this without unnecessary programming? SAP HR provides “features” for this purpose. I’ll show you how to use this tool next.

**Using HR “features”**

Features are special HR objects that determine a specific value by querying HR fields (e.g., infotypes or system structures). The return code of the feature (usually the value) is used to control HR processes or to set default values for infotypes. You can use standard features as is or modify them, or you can create your own.

Features are maintained using a special editor (transaction PE03, HR Features) and must be “generated” (this process is also sometimes referred to as “activation”). Feature generation results in a report or value in table T549D that can be used by HR programs. Most features — e.g., ABKRS, SCHKZ, TARIF — are used to determine the default values for HR infotypes when the infotypes are maintained. Other features — e.g., Pnmm features (P0004, P0006, etc.) — are used to control infotypes’ single screens. Every feature is assigned a structure with decision

![Image of enhanced single screen for infotype 0006](Image)

**Figure 12**  Enhanced single screen for infotype 0006
fields that can be used to modify the feature. The example structure for standard feature ABKRS is shown in Figure 13.

When the user runs the Hiring personnel action, for example, some infotype fields may get default values according to different rules (e.g., according to employee group or personnel area assignment). Feature ABKRS is used to generate the default value for the Payroll area field in the Organizational Assignment infotype (P0001-ABKRS). Figure 14 shows a sample decision tree for feature ABKRS.

Let’s read a fragment of the ABKRS tree. When the transaction class (TCLAS) is A (Master data and time data), the country grouping (MOLGA) is 10 (USA), and the employee subgroup (PERSK) is U1, the default value for the payroll area (ABKRS) should therefore be UW.

Editing a feature tree is easy — just use transaction PE03. Remember that each feature must be activated to be used. The result is an ABAP feature program and an entry in table T549D (Feature Directory). The feature program is not generated if the decision tree contains only a return value (which doesn’t exceed

**Note!**

Some standard infotypes cannot be enhanced (there is no CI_Pnnnn include). For example:

- Actions (infotype 0000)
- Additional Actions (infotype 0302)
- Time Management (infotypes with the number 2nnn, such as 2001, 2002, 2003, etc.)
- Applicant Actions (infotype 4000)

![Figure 13](Structure for standard feature ABKRS)
For such features, the FUNID field in table T549D contains the entry V (for value, rather than P for program) and the return value is stored in the FUNCT field (in the same table). For additional technical information about HR features, check SAP Note 321714.

Let’s create a new feature for our enhanced Addresses screen (Figure 12):

1. The easiest way to create a new feature is to copy an existing feature and modify it. So, run transaction PE03, enter the feature name P0006, click on Copy ( ), and enter a new feature name (Z0006), as shown in Figure 15. Select Decision tree and click on Change ( ).
2. Using the Delete node/sub-tree ( ) and Create nodes ( ) functions, prepare the decision tree as shown in Figure 16. As you can see, according to this feature, when the country grouping (MOLGA) is something other than 08 (Great Britain) and the subtype (SUBTY) is 1 (Permanent residence), the feature return value is 56; otherwise it is a “space.”

3. To use the feature with our enhanced infotype, we need to control screen 0200 (remember that the fields we added to the infotype are defined in screen 0200 of module pool ZP000600) using table T588M, which is used for modifying infotype screens according to the rules defined in features. Run transaction SM30 (Extended Table Maintenance), enter the table name (T588M), and click on the Maintain button. Click on New entries and enter the data shown in Figure 17. Now copy the new entry, empty the variable key field, and set all fields as hidden.

4. To check the results, run transaction PA30 and try to create a new record for infotype 0006, subtype 1. You will see the subscreen with additional fields. When you try to create a record with any subtype other than 1, the subscreen will be invisible.

As you can see, HR features are a useful way to modify the behavior of infotypes. But the available features and their functionality are limited, and sometimes you will need to make more substantial changes to infotype processing logic than features can support. For these cases, SAP provides a more flexible tool, HR function module exits.

Using HR function module exits

SAP provides predelivered enhancements containing exits that allow you to modify standard SAP code. The PBAS0001 enhancement is the most frequently used enhancement in SAP HR projects. You can use the exits it contains in the Process Before Output (PBO) or Process After Input (PAI) of an infotype to modify its behavior.4 The PBAS0001 enhancement consists of two function module exits:

- EXIT_SAPFP50M_001 — The function module called in the PBO when any infotype data record is created

4 Instead of using PBAS0001, you can implement BAId HRPAD00INFETY. See the documentation for transaction SE18 for more information.
• EXIT_SAPFP50M_002 — The function module called in the PAI when any infotype record is created or edited

Let’s see how it works with EXIT_SAPFP50M_001, which is usually used to set default values for infotype screen fields. Suppose we want to set a default value for the Address type field in infotype 0006 whenever a new address record is created for the employee assigned to subgroup DN (the subgroup assignment is kept in infotype 0001). We do this by inserting the code shown in Figure 18 in include
As you can see, structure ipsyst (defined by Data Dictionary structure PSYST) is used to check the employee subgroup (ipsyst-persk). This structure interface contains many other organizational assignment fields for the employee, such as Company Code, Business Area, Personnel Area, Employee Group, Payroll Area, and Cost Center. Check the PSYST structure in the Data Dictionary for available HR fields.

Note!
In Unicode programs, pnnnn must have the same structure layout as innnn, independent of the length of a Unicode character (see SAP Note 562145). SAP recommends converting transfer parameter innnn to the infotype structure (methods of class cl_hr_pnnnn_type_cast will carry out this conversion).

Warning!
Values from EXIT_SAPFP50M_001 are used only while creating new records by inserting, not when copying preexisting values.

Figure 18 (continued)

prelp = innnn
importing
  pnnnn = i0006.

if ipsyst-persk eq 'DN'.
  move '1' to i0006-anssa.
endif.

call method cl_hr_pnnnn_type_cast=>pnnnn_to_prelp
  exporting
    pnnnn = i0006
  importing
    prelp = innnn.

  when others.
  endcase.

Now you can check the results of your work. When you create a new record of infotype 0006 for the employee belonging to subgroup DN (in transaction PA30), the default address type is 1 (Permanent Residence).

Note!
For some infotype fields, it is possible to use a parameter ID (check if the field in question has a parameter ID in that field’s technical information).
You have now learned how to use HR features to make simple infotype behavior modifications without any coding required, and how to use HR function module exits in the PBO and PAI of infotypes for more substantial changes to the infotype processing logic. Sometimes a standard SAP-provided infotype isn’t enough, however — even with enhancements. In these cases, you need to create a custom infotype to meet your specific needs.

Creating custom infotypes

Suppose you require a new infotype for maintaining data about internal training. The infotype should contain the following fields:

- Start date (beginning date for the training)
- End date (end date for the training)
- Training type (a check table should be defined)
- Training description line 1
- Training description line 2

Let’s create this infotype:

1. Run transaction SE11 (ABAP Dictionary) and create a table (ZTRAIN) with fields, as shown in Figure 19 on the next page. Create a few sample records in the table, as shown in Figure 20 on the next page.
Figure 19  Dictionary table ZTRAIN for training

Figure 20  Sample ZTRAIN records
2. Run transaction PM01, enter 9000 for the new infotype number (9000 is the first free number for customer infotypes, but you can use any value in the range 9000–9999), select PS structure, and click on Create (see **Figure 21**). Define the PS9000 structure as shown in **Figure 22** and activate it.

It is a good idea to also create an elementary search help ZTRAIN in transaction SE11 with the

---

**Figure 21**  PM01 screen for infotype creation

**Figure 22**  PS9000 structure
selection method ZTRAIN and the export parameter ZTTYP. Remember to assign the check table and search help ZTRAIN to the component ZZTYP.

3. Return to the PM01 screen and click on Create All ( ) to display the “Infotype attributes” view. Infotypes have several characteristics (e.g., time constraints), which are defined in table T582A. Now create a new record for our infotype in this table. The easiest way to do this is to copy the record of a standard infotype that is similar in behavior to the new infotype. For our example training infotype, let’s copy the record for infotype 0022 (Education), which gives us the screen shown in Figure 23. Save your entry.

4. To make our infotype user friendly, let’s add the training type description text (ZTRAIN-ZTEXT) in the field to the right of the type identifier. We can do this with transaction PM01 or SE80 (Object Navigator). I suggest using SE80 since it provides better navigation capabilities than PM01. In transaction SE80, choose object Program and object name MP900000 (the module pool generated for our infotype).

Figure 23   Infotype 9000 attributes
- First we create the table work area for ZTRAIN in the MP900010 include that is dedicated to data definitions (Figure 24).

- Next we change the generated PBO module P9000 to ensure that the training type description text will be read whenever the single screen is shown. We insert the call of procedure RE_ZTRAIN in module P9000, as shown in the first line of the code in Figure 25.

Figure 24 Creating a table work area for ZTRAIN in include MP900010

Figure 25 Inserting the procedure call in PBO module P9000

Continues on next page
IF PSYST-NSELG EQ YES.
* read text fields etc.; do this whenever the screen is show for the
* first time:
*   PERFORM RExxxx.

   IF PSYST-IINIT = YES AND PSYST-IOPER = INSERT.
* generate default values; do this the very first time on insert only:
*     PERFORM GET_DEFAULT.
   ENDIF.
ENDIF.
ENDMODULE.

*----------------------------------------------------------------------*
*       MODULE  P9000L OUTPUT                                          *
*----------------------------------------------------------------------*

*       read texts for listscreen
*----------------------------------------------------------------------*

MODULE P9000L OUTPUT.
* PERFORM RExxxx.
ENDMODULE

*----------------------------------------------------------------------*
*                                                                      *
*       Subroutines for infotype 9000                                 *
*                                                                      *
*----------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*&      Form  RE_ZTRAIN                                              *
*&---------------------------------------------------------------------*
*       Reading text for training type                              *
*----------------------------------------------------------------------*
form RE_ZTRAIN.

select single * from ZTRAIN where SPRSL eq SY-LANGU
   and ZTYP eq P9000-ZTYP.
if sy-subrc ne 0.
clear ZTRAIN.
endif.
endform.                    " RE_ZTRAIN

Figure 26  Procedure RE_ZTRAIN definition
- Define procedure RE_ZTRAIN in include MP900040, as shown in Figure 26.

- Lastly, place the ZTRAIN-ZTEXT field in screen 2000 using the Screen Painter. Choose screen 2000 and insert a new Input/Output field with the name ZTRAIN-ZTEXT, as shown in Figure 27. Confirm that the field is taken from the ABAP Dictionary.

Once you have created a custom infotype, you can use HR features and HR exits to enhance it as you would for a standard infotype.

That’s it — you now know how to enhance standard infotypes and create custom ones. While I focused on PA infotypes in this article, the steps are similar for developing other kinds of infotypes as well. In the case of OM infotypes, however, you use a different transaction. For more on this, see the download available at www.SAPpro.com.

Before users start maintaining records based on your enhanced and custom-created infotypes, there is one more useful tool I would like to share that can help simplify this task for users. Nearly every SAP HR implementation project has at least one
requirement to initiate a specific action (e.g., sending mail or calling a procedure) when maintaining a particular PA infotype record. You can, of course, hard code the required logic, but the PA component provides a better way to handle such requirements — dynamic actions. In this last section, I’ll show you how to implement this feature to streamline infotype data maintenance for your users.

Implementing dynamic actions

A dynamic action automatically initiates an activity based on the maintenance of the infotype data.

Tip!

You can control a screen using modification groups 1 and 3. The meaning of modification group 1 is defined in table T589A as shown below:

<table>
<thead>
<tr>
<th>Hexadecimal constant for modification group 1</th>
<th>Mode in which entries can be made in screen fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Display</td>
</tr>
<tr>
<td>002</td>
<td>Change</td>
</tr>
<tr>
<td>004</td>
<td>Add and Copy</td>
</tr>
<tr>
<td>008</td>
<td>Delete</td>
</tr>
<tr>
<td>010</td>
<td>Lock/unlock</td>
</tr>
</tbody>
</table>

The values shown in the first column may be OR’ed — e.g., 006 means that for a given field the entry can be made when the record of the infotype is being created (added), copied, or changed.

Modification group 3 is a much stronger tool for controlling screen behavior. The value of modification group 3 (between 001 and 050) for each field of the infotype lets you modify the behavior of this field via table T588M and an HR feature (such as the feature shown earlier in the example infotype 0006 enhancement). For example, you can set the field as required, optional, or hidden. Usually HR features used here include the name of the structure for the infotype (e.g., the feature controlling screen behavior for infotype 0002 has the name P0002).

In Figure 27, you can see sample modification group settings for the field ZTRAIN-ZTEXT (group 1 = 006, group 3 = 001).

A dynamic action may be called when you change, insert, or delete an infotype record. It can be carried out if the user changes specific fields or any field of a chosen infotype record. The system lets you define the additional conditions for executing the action. Here are a couple of examples:

- The system should generate, in the background, a new record of infotype 0019 (Monitoring of Tasks) whenever an infotype 0016 (Contract elements) record is changed or created. Infotype 0019 keeps data needed for automatic monitoring of HR events such as expiry of probation, expiry of temporary contract, or next appraisal. If, for example, an employee is hired for the probation period
No portion of this publication may be reproduced without written consent.

SAP HR programming demystified: Understanding, creating, and enhancing SAP HR infotypes

Figure 28  Sample dynamic action definition

(information in infotype 0016), the deadline date for changing the contract should be stored in infotype 0014.

- The system should send mail to the HR administrator whenever a record of infotype 0001 is changed.

Dynamic actions are defined in table T588Z. Let’s look at the dynamic action for the following example: Whenever the user creates a record for wage type M017, the system should enforce wage type M018 maintenance (note that these wage type codes are for example purposes only — they have no special meaning).

Run transaction SM31 (Standard Table Maintenance) to maintain table T588Z. The maintenance is easy — you can insert, copy, and delete entries. Insert the lines shown in Figure 28.

Figure 29 on the next page describes the meaning of each column in the view of T588Z.

Let’s analyze the dynamic action defined in Figure 28 line by line:

1. If the user creates (FC = 04) a record of infotype 0015 (IType) with the country grouping (MOLGA) 46 (Poland)…
2. And if the wage type (P0015-LGART) in the new record has the code M017…
<table>
<thead>
<tr>
<th>Column</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IType (Infotype number)</td>
<td>Defines the infotype for which the action should be performed.</td>
</tr>
<tr>
<td>STy. (Subtype)</td>
<td>Defines the subtype for which the action should be performed. You use this column when you want an action invoked for a specific subtype, but not all records.</td>
</tr>
<tr>
<td>Field N (Field name)</td>
<td>Identifies a specific infotype field for which you want an action performed.</td>
</tr>
<tr>
<td>FC (Function character of step)</td>
<td>Defines the processing type for which a dynamic action should be performed. The following processing types are available:</td>
</tr>
<tr>
<td></td>
<td>• 02: Change</td>
</tr>
<tr>
<td></td>
<td>• 04: Create</td>
</tr>
<tr>
<td></td>
<td>• 06: Change and create</td>
</tr>
<tr>
<td></td>
<td>• 08: Delete</td>
</tr>
<tr>
<td></td>
<td>• 10: Change and delete</td>
</tr>
<tr>
<td></td>
<td>• 12: Create and delete</td>
</tr>
<tr>
<td></td>
<td>• 00: Independent of the function being performed</td>
</tr>
<tr>
<td></td>
<td>For example when FC = 04, the action is performed when the specified (in column IType) infotype record is created.</td>
</tr>
<tr>
<td>No (Sequence number)</td>
<td>Defines the sequence in which the dynamic action code is executed.</td>
</tr>
<tr>
<td>S (Indicator of step)</td>
<td>Defines the type of operation used in the variable function part. Variants include:</td>
</tr>
<tr>
<td></td>
<td>• P: Plausibility checks</td>
</tr>
<tr>
<td></td>
<td>• I: Maintain infotype record</td>
</tr>
<tr>
<td></td>
<td>• W: Set default values when creating a new record</td>
</tr>
<tr>
<td></td>
<td>• V: Reference to another step</td>
</tr>
<tr>
<td></td>
<td>• F: Call routine</td>
</tr>
<tr>
<td></td>
<td>• M: Send mail</td>
</tr>
<tr>
<td></td>
<td>You can also enter comments that you want to appear in a variable function part.</td>
</tr>
<tr>
<td>Variable function part</td>
<td>Further defines the operation that is defined in column S. For example, if S=I, the possible actions are:</td>
</tr>
<tr>
<td></td>
<td>• INS: Create a record</td>
</tr>
<tr>
<td></td>
<td>• COP: Copy a record</td>
</tr>
<tr>
<td></td>
<td>• MOD: Modify an existing record</td>
</tr>
<tr>
<td></td>
<td>• DEL: Delete a record</td>
</tr>
</tbody>
</table>

**Figure 29**  T588Z column headers

3. Create a new record of infotype 0015 with subtype M018 and the same start date (P0015-BEGDA) as the record for M017…

4. Finally, put the string ‘WAGE TYPE M017’ into the assignment number field (P0015-ZUORD) of a new record.
If you want to check the results of your work, create a new record of infotype 0015 for subtype M017 (using transaction PA30).

Dynamic actions are often used in personnel actions. For a new employee, some actions should be performed automatically (e.g., creating specific infotype records in the background or sending an email to an administrator). The following are examples of dynamic actions you can use in your own projects:

- **Delete a record of infotype 0015, subtype M117, initiated by the start date of another record of infotype 0015:**
  - **IType:** 0015
  - **FC:** 08
  - **No:** 1
  - **Variable function part:**
    
    DEL,0015,M117,,,(P0015-BEGDA)

- **Execute a procedure GET_DATE defined in report HBRDYNMS:**
  - **IType:** 0000
  - **FC:** 04
  - **No:** 23
  - **Variable function part:**
    
    GET_DATE(HBRDYNMS)

- **Send a mail with characteristics defined by feature M0001 (see transaction PE03 for details):**
  - **IType:** 0001
  - **Field N:** SACHP
  - **FC:** 06
  - **No:** 1
  - **Variable function part:** M0001

Dynamic actions can be helpful tools that let you avoid a lot of coding or even repair of standard programs. Unfortunately, however, dynamic actions are not syntax checked, so you should be very careful while editing the variable function part. The examples of dynamic actions in a standard system (see table T588Z) will serve as a valuable source of further information regarding the effective use of dynamic actions.

### Note!

Remember that in table T588Z, there are no checks performed on the data, and therefore correct spelling is very important.

### Tip!

Dynamic action definition can make use of the table strings PSPAR, T001P, and T503 valid on the current infotype record start date. Additionally, the structure PSAVE contains the old values of the infotype record. For example, if some actions should be taken only when an employee belongs to the specific personnel area (an important employee attribute in SAP HR) you can check the T001P-WERKS field (personnel area) in your dynamic action. The field contains the personnel area for a given employee. In the dynamic action analyzed in the example in Figure 28, the field T500P-MOLGA (country grouping) is checked.

### Conclusion

This article certainly hasn’t covered all facets of infotype programming, but has focused instead on only the crucial aspects: infotype creation and enhancement, HR features, HR function module exits, and dynamic actions — these are the core areas where added knowledge will help you cope with most HR project challenges.
SAP provides several tools (e.g., transactions PM01, PE03, and CMOD) that will make your work with infotypes more effective through the use of automatic code generation, configuration instead coding, and enhancements instead of repairs. Using them will help you avoid modifications to standard SAP code and reduce the risk of problems during upgrades. I have tried to condense here for you a set of practical knowledge, choosing those HR issues that occur frequently in all SAP HR projects and are release-independent. It’s my hope that this level of generality will aid you in implementing your own successful HR projects!