

Features Editor



HELP.PAXX

Release 4.6C



Copyright

© Copyright 2001 SAP AG. All rights reserved.

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

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

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] and SQL Server[®] are registered trademarks of Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®], and OS/400[®] are registered trademarks of IBM Corporation.

ORACLE[®] is a registered trademark of ORACLE Corporation.

INFORMIX[®]-OnLine for SAP and Informix[®] Dynamic Server[™] are registered trademarks of Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®], and Motif[®] are registered trademarks of the Open Group.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT[®] is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Contents

Features Editor	6
Feature.....	7
Calling Feature Maintenance	9
Persons Responsible for Features	10
Country/Component Assignment for a Feature.....	11
Feature Structure.....	12
Feature Structure Maintenance.....	13
Feature Documentation.....	15
Decision Tree for a Feature.....	16
Decision Tree in Tree Maintenance	17
Nodes in the Decision Tree (Tree Maintenance).....	20
Return Operation/Tree Maintenance	21
Decision Operation/Tree Maintenance	22
Subfeature Operation/Tree Maintenance	25
Program Operation/Tree Maintenance	26
Debugging Operation/Tree Maintenance.....	27
Error Operation/Tree Maintenance	28
Decision Tree in Table Maintenance	29
Return Operation/Table Maintenance.....	30
Decision Operation/Table Maintenance.....	31
Follow-On Operation/Table Maintenance	33
Subfeature Operation/Table Maintenance.....	34
Program Operation/Table Maintenance.....	35
Debugging Operation/Table Maintenance	36
Error Operation/Table Maintenance	37
Additional Information on Program Operations.....	38
Editor Commands	41
Line Commands	42
The Decision Tree of Feature LVNUM Example.....	43
Technical Information: Reading Features in Program	45
Maintaining Features	47
Creating a Feature	49
Creating the Persons Responsible for a Feature.....	50
Creating the Country/Component Assignment for a Feature.....	51
Creating a Structure	52
Creating a Structure for a New Feature	53
Creating Documentation	54
Creating the Decision Tree for a Feature.....	55
Creating the Decision Tree for a Feature (Tree Maintenance).....	56
Creating the Decision Tree for a Feature (Table Maintenance)	58
Changing a Feature.....	60
Changing the Persons Responsible for a Feature.....	61
Changing the Country/Component Assignment for a Feature.....	62
Changing the Structure of a Feature.....	63

Changing Documentation.....	64
Changing the Decision Tree of a Feature	65
Technical Information: Changing a Standard Feature	66
Copy a Feature.....	67
Copying a Feature.....	68
Activate a Feature	69
Activation Types.....	70
Activating a Feature	71
Technical Information: Activating a Feature.....	72
Deletion of a Feature	74
Deleting Features.....	75
Environment for Maintaining Features	76
Technical Information: Saving a Feature	77
Utilities and Tools	78
Creating a Feature Directory	80
Comparing Features.....	81

Features Editor

Use

You can edit features with the Features Editor

Integration

Features contain rules that are primarily used for programming in the HR component and that are, for the most part, used for specifying default values.

Feature

Definition

Features are objects in the SAP system, which by querying different enterprise personnel or data structure fields, determine a specific value. The value is known as a “return code” or “result”. This value is used to determine default values and to control certain system processes. Features serve to enhance system flexibility.

Use

Features can be queried in programs. To make this possible, every feature must be activated. A report is created from the feature in the activation process. Programs that require feature information utilize this report.

The most important uses of features are outlined below:

- **Features for determining default values**

Features are frequently used to determine default values. These values are proposed by the system when employee master data infotypes are maintained.

Feature ABKRS, for example, provides a default value for the payroll group. This is based upon the information found in an employee’s *Organizational Assignment* infotype. Feature LGMST also illustrates this concept: This feature proposes the valid wage and salary types for an employee, which are taken from the *Basic Pay* infotype. The feature also refers to the personnel and pay scale structure.

- **Features for controlling system processes**

Pnnnn features (nnnn is → infotype number) control the sequence of country-specific screens for displaying and maintaining master data infotypes. For example, if you set the country grouping as the user group parameter (e.g. UGR=01), the system references the country-specific screens.

- **Features for controlling list displays in evaluations**

Feature BUWEB, for example, can be used to group or convert personnel areas or personnel subareas for evaluations. This enables you to change the sort sequence of lists as required.

Structure

Features are defined by the following five components, which must be maintained in the order listed:

- [Persons Responsible for a Feature \[Page 10\]](#)
- [Country/Component Assignment for a Feature \[Page 11\]](#)
- [Feature Structure \[Page 12\]](#)
- [Decision Tree for a Feature \[Page 16\]](#)
- [Feature Documentation \[Page 15\]](#) (optional)

Feature

See also:

[Environment for Maintaining Features \[Page 76\]](#)

Calling Feature Maintenance

Use

You can access the maintenance functions for the decision tree for a feature in different ways.

Prerequisites

You are in one of the following places in the SAP System:

- Implementation Guide (IMG) for *Personnel Management*
- SAP initial screen *SAP Easy Access*
- Decision tree for a feature

Procedure

If you are in the Implementation Guide (IMG) for <i>Personnel Management</i> ,	call the IMG step to which the feature is assigned
If you are in the SAP initial screen <i>SAP Easy Access</i> ,	you access feature maintenance directly with transaction PE03
If you are in the decision tree for a feature,	you can maintain another feature by choosing <i>Feature</i> → <i>Other feature</i>



If you want to change the structure, country/component assignment, persons responsible, or the feature documentation, choose *Goto* in the decision tree for the feature.

Result

You are in the initial screen of the feature maintenance.



All procedures for feature maintenance are documented from the initial screen *Features: Initial Screen*.

See also:

[Environment for Maintaining Features \[Page 76\]](#)

Persons Responsible for Features

Persons Responsible for Features

Definition

Unlike schemas and personnel calculation rules where a single responsible person is defined for maintenance, a feature allows for the assignment of two responsible persons.

Use

One person is assigned responsibility for maintaining the [Feature Structure](#) while another is assigned responsibility for the feature as a whole, including the following.

- [Country/Component Assignment for a Feature \[Page 11\]](#)
- [Decision Tree for a Feature \[Page 16\]](#)
- [Feature Documentation \[Page 15\]](#)

You can specify whether assigned users are the only persons allowed to make changes to their respective parts of the feature.

Note that changes to the structure of a feature can also necessitate changes to the coding of the program behind it. Users responsible for changing structures must have the authority and technical ability to be able to maintain dictionaries and make program changes.



You do not have to specify two different users as the persons responsible. One individual may also be responsible for the entire feature.

In the *Title* field, you can create and edit a short description of the feature. The title then appears next to the feature field on every screen.

This screen also contains a section entitled *Administrative data*. The information contained in this section defines the status of the feature. The status covers:

- the creation date
- the date on which the feature was last changed (and the person responsible for changing it)
- the version (number of changes implemented in the meantime)
- the status (*saved* = 1 or *stored temporarily* = 2 or 4)
- the name of the generated program

See also:



[Create Persons Responsible \[Page 50\]](#)

Country/Component Assignment for a Feature

Definition

The country assignment indicates the countries to which the feature applies. The component assignment determines how and where the feature is used in the HR system.



The country assignment is an *attribute* of a feature. You can find the country/component assignment on the initial screen of Attribute Maintenance. Under *Subobjects*, select the field *Attributes* and choose  *Display* or  *Change*. The system displays the screen *Display or Change Feature xxxxx : Attributes*, where xxxxx is the name of the feature. Here you choose *Country/Component Assignment*.

Use

When you create a feature, you are required to specify at least one country assignment and at least one component assignment.

For example, you can define a feature for all countries and use it within incentive wages and group incentive wages.

See also:

[Creating the Country/Component Assignment for a Feature \[Page 51\]](#)



Feature Structure

Feature Structure

Definition

A structure for decision fields must be assigned to every feature. A structure consists of one or more fields.



The structure is an *attribute* of a feature. You can find the structure on the initial screen of Attribute Maintenance. Under *Subobjects*, select the field *Attributes* and choose  *Display* or  *Change*. The system displays the screen *Display or Change Feature xxxxx : Attributes*, where xxxxx is the name of the feature. Here you choose Structure.

Use

As the person responsible for a feature, choose an existing structure depending on the feature functions and required fields, or create a structure specifically for this feature. Only the structure fields that you mark are recognized by the programs that read the features. That is why only the marked fields can be used in the decision tree.



Note that changes to the structure of a feature can affect the coding of the programs that read such features. You can find further information in [Persons Responsible for Features \[Page 10\]](#).

You can maintain feature structures. You can find further information in [Maintaining the Structure of a Feature \[Page 13\]](#).

See also:

[Creating a Structure \[Page 52\]](#)

Feature Structure Maintenance

Purpose

You can maintain the structure that you have created for a feature.

Prerequisites

The following entries belong to structure maintenance:

- Structure name for decision fields
- Structure name for back value
- Structure name for back value
- Transfer type
- Subfeature
- Valid field names for decisions

Process flow

- **Structure name for decision fields**

Structures are defined in the ABAP Dictionary. Only field names defined in the structure can be used in the decision tree. For further information on the ABAP Dictionary, please refer to the documentation on the **ABAP Workbench** under **BC ABAP Dictionary** in the R/3 Library.

- **Back value structure name**

If the back value consists of more than one value for field return or of more than one column for table return, you can enter the name of a suitable structure here.

The system proposes the individual fields of this structure when you choose the *create* or *change* functions in the decision tree. If the fields have a check table or fixed values, these values are displayed as input help, and the entries are checked against these values.



For technical reasons, the structure of the back value can have a maximum size of 42 bytes. You can enter either a structure name or field name for the back value.

- **Back value field name**

If the field has a check table or refers to a domain with fixed values, the system will propose the key values of the check table or the fixed values as possible back values. This occurs when you are using the create or change functions in the decision tree.



You must enter the back value in the form *Table name - Field name*, for example, *P0001-ABKRS*. You can enter either a structure name or field name for the back value.

- **Transfer type**

Feature Structure Maintenance

The transfer type determines whether the back value should be a field transfer (1) or a table transfer (2).

- **Subfeature**

If this field is flagged, the feature is created as a subfeature. The subfeature is included in the main feature. You can find further information in [Decision Tree for a Feature \[Page 16\]](#).

- **Valid field names for decisions**

As the person responsible for the feature, select the fields that can be used in the decision tree for the feature. You select these from the fields defined for the feature structure.



Please note that you do not have to use all of the fields available.

You can find further information on feature maintenance in [Environment for Maintaining Features \[Page 76\]](#).

Feature Documentation

Definition

Feature documentation describes a feature's various functions.

Use

You can write documentation to describe a feature. It is client-specific and language-dependent.

See also:

[Creating Feature Documentation \[Page 54\]](#)

Decision Tree for a Feature

Decision Tree for a Feature

Definition

The decision tree of a feature contains rules specifying the back value or return table required to execute certain functions within the HR component.

Use

You can use a table editor and, as of release 4.5A, a maintenance tree, to maintain these rules.

Decision tree reporting can be reduced to the following basic form:

- Enter *back* as the back value/table (back value operation).
- If the */1* appears in field *f1*.....enter back as the back value/table (decision operation).

A range of operations are available to help you structure complex decision trees and for special functions. Because both the representation and implementation of these operations are completely different in the maintenance tree and the table editor, they are described separately.

[Decision Tree in Tree Maintenance \[Page 29\]](#)

[Decision Tree in Table Maintenance \[Page 29\]](#)

Decision Tree in Tree Maintenance

Use

Operations control processing within the decision tree.

You can execute the following operations in the decision tree:

- [Return Operation/Tree Maintenance \[Page 21\]](#)
- [Decision Operation/Tree Maintenance \[Page 22\]](#)
- [Subfeature Operation/Tree Maintenance \[Page 25\]](#)
- [Program Operation/Tree Maintenance \[Page 26\]](#)
- [Debugging Operation/Tree Maintenance \[Page 27\]](#)
- [Error Operation/Tree Maintenance \[Page 28\]](#)

Features

You can use the following functions for tree maintenance:

- *Create*

To use this function, you must position the cursor on a node.



You can find further information on nodes in [Nodes in the Decision Tree \[Page 20\]](#).

With the *Create* function, you insert a new node into the decision tree at the next lowest hierarchy level.

Exception 1: Creating on Field Value Type Node: In this case, it is possible to create a commentary that will be inserted in the same hierarchy level.

Exception 2: If you create a decision-operation type node, then an “otherwise” node will be simultaneously created directly under it, and should be displayed at all times.

- *Change*

Position the cursor on a node in the decision tree.

In general, the content of the node is displayed for change in a dialog box. Exceptions to this are situations where the system suggests possible entries. In these cases, the system offers you the following possibilities:

- *Create Interval Value*

See *Create* tree function.

The function *Create interval value* is intended for situations in which the system has displayed an interval of possible entries, and for when you want to create one of these interval values.

- *Change Interval Value*

See *Change* tree function.

Decision Tree in Tree Maintenance

The function *Change interval value* is intended for situations in which the system has displayed an interval of possible entries, and for when you want to select the new value from the interval values displayed.

- *Check*

This tree function checks whether the decision tree contains permissible operations. The error messages are displayed in the status bar. If you click on this message bar, you can access the long text message. The nodes with errors are marked in red.

- *Save*

This tree function causes the current decision tree to be checked and eventually saved. You can further display and edit the decision tree both in tree maintenance and table maintenance.

- *Save Without Checking*

This tree function causes the current decision tree to be checked and eventually saved. You can display and edit the decision tree in tree maintenance only.

- *Activate*

This tree function contains the tree function *Save*. After saving, a program is created from the decision tree. This program is read by the programs that require feature information.

The *Activate* function, initiated on the *Features: Initial Screen* (transaction PE03), uses the last saved and checked version to generate the feature program.

The tree function *Activate*, initiated in the decision tree, generates the feature program from the current decision tree.

- *Choose*

This tree function is possible for the following node types:

- Decision-operation type nodes: field name
- Subfeature call
- Program call

The tree function navigates to the relevant objects. For technical reasons, navigating to a subfeature is only possible in the display mode.

- *Delete*

Position the cursor on a node in the decision tree.



Not only will the node on which you have positioned the cursor be deleted but so too will all the nodes that are dependent on this node, that is, the relevant sub-tree.

- *Select/Deselect*

Position the cursor on a node in the decision tree.

If the tree functions *Cut* and *Copy* are to apply to more than one node, then you must select the nodes.

- *Select Block*

Decision Tree in Tree Maintenance

Position the cursor on the node that represents the beginning of the block you want to select. Then, position the cursor on the node that represents the end of the block.

The *Select block* function is necessary for the *Cut* and *Copy* functions if you want to select more than one node, or do not want to individually select every single line.

- *Cut*

Position the cursor on a node in the decision tree.



Not only will the node on which you have positioned the cursor be cut and saved in an internal buffer, but so too will all the nodes that are dependent on this node, that is, the relevant sub-tree.

- *Copy*

Position the cursor on a node in the decision tree.



Not only will the node on which you have positioned the cursor be copied and saved in an internal buffer, but so too will all the nodes that are dependent on this node, that is, the relevant sub-tree.

- *Paste*

Position the cursor on a node in the decision tree.

The internal buffer, which was filled through the *Cut* or *Copy* actions, is copied to the nodes marked by the cursor. A dialog box appears asking you whether the contents of the buffer are to be copied at the same level or subordinately.

- *Move*

Before you select the *Move* function, first mark the nodes or sub-trees to be moved.

The marked nodes/sub-trees are appended to the node where your cursor is positioned. A dialog box appears, asking you whether the nodes are to be appended at the same level or subordinately.

Nodes in the Decision Tree (Tree Maintenance)

Nodes in the Decision Tree (Tree Maintenance)

Definition

Accessing the decision tree with tree maintenance.

Use

Nodes have different meanings, represented by different color legends. You can access the color legends and an overview of the node types via *View* → *Color legend*.

The root node always represents the feature in question. The following types of nodes are positioned under the root node and the field value nodes:

- Return value
- Decision operation field
- Subfeature call
- Program call
- Debugging operation
- Error operation

Return Operation/Tree Maintenance

Use

Return operations define the back value of the feature.

Decision Operation/Tree Maintenance

Decision Operation/Tree Maintenance

Use

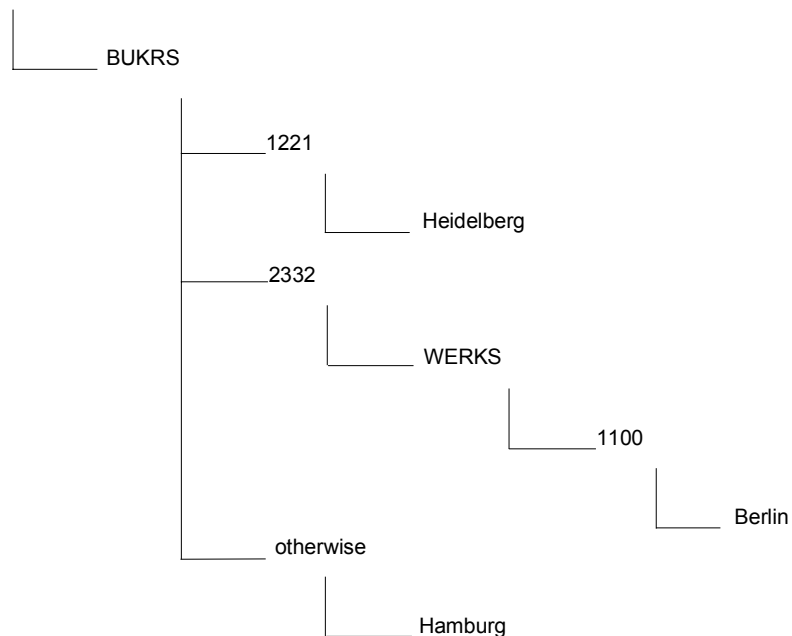
To execute a decision operation, select a field from the feature structure suggested by the system.

- **Variant 1:**

You want to specify different return codes for specific fields in the decision tree.



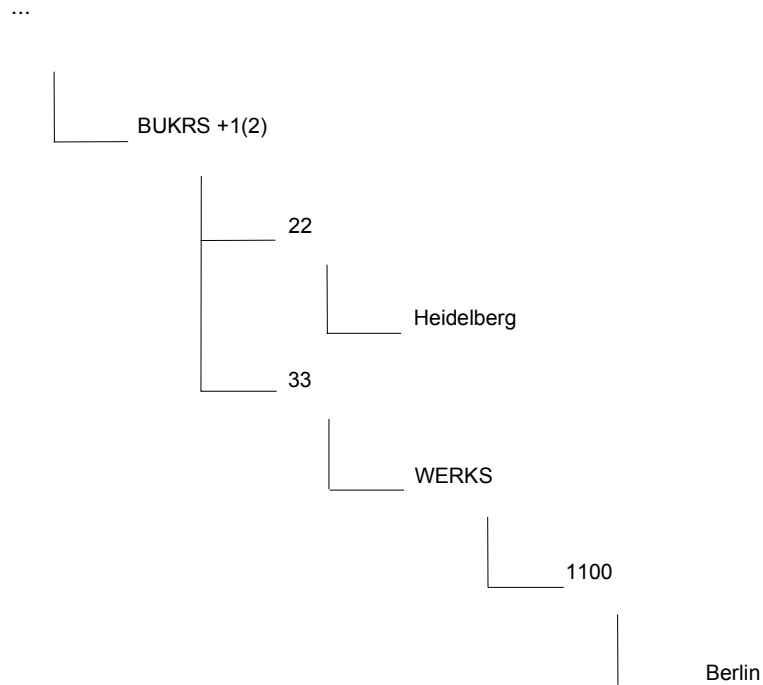
...



- **Variant 22:**

You do not want to analyze the decision operation fields in their full length. If you create the field with offset and length, the follow-on nodes have the relevant shorter content.





- **Variant 3:**

In the decision operation, the field from the feature structure is compared with a value or another field in the structure. In this case, the follow-on node is a comparison operator. Operators that can be used here are:

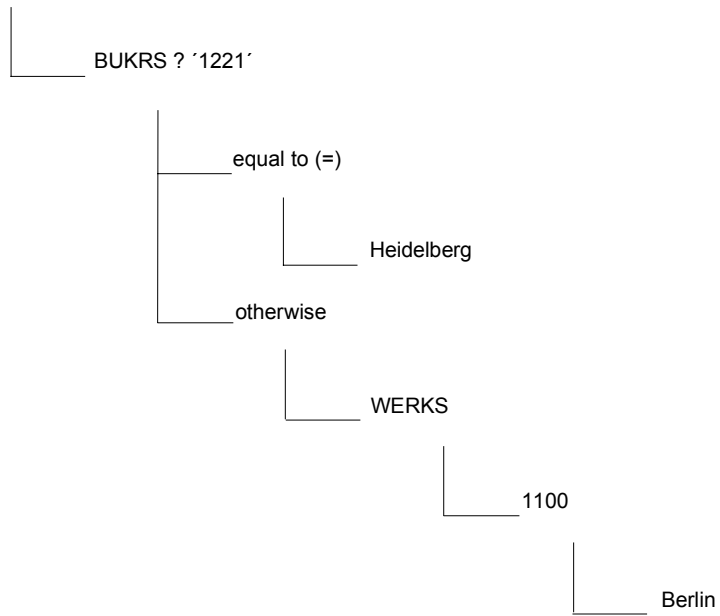
- > (bigger than)
- < (smaller than)
- = (equal to)

You can also use * (other) for unspecified cases.



Decision Operation/Tree Maintenance

...



Subfeature Operation/Tree Maintenance

Use

You can call up a subfeature with the subfeature operation. Processing is continued in this subfeature. It is advisable to use this operation if several features can use the same decision tree. This is then created as a subfeature.

Program Operation/Tree Maintenance

Use

The program operation offers you the possibility of not explicitly entering the back value of a feature in the decision tree. You can let the ABAP program define the back value.

See also:

[Additional Information on Program Operation \[Page 38\]](#)

Debugging Operation/Tree Maintenance

Use

You use the BREAK operation to access the user-dependent debugging mode for troubleshooting purposes.

You must enter a valid user name here.



Please note that the debugging operation is always entered additionally to the decision- or return operation.

Error Operation/Tree Maintenance

Use

Operation ERROR does not transmit any back value. Instead, error status 2 is set. Error status 2 displays a constellation that is not permitted in the decision tree.

Decision Tree in Table Maintenance

Use

The decision tree consists of the following elements:

- **Operations**
 - Operations control processing within the decision tree.
 - The following operation types can be distinguished:
 - [Return Operation \[Page 30\]](#)
 - [Decision Operation \[Page 31\]](#)
 - [Follow-Up Operation \[Page 33\]](#)
 - [Subfeature Operation \[Page 34\]](#)
 - [Program Operation \[Page 35\]](#)
 - [Debugging Operation \[Page 36\]](#)
 - [Error Operation \[Page 37\]](#)
 - **Rule type (under heading D)**
 - Rule type *D*
 - This rule type is used to define decision operations.
 - Rule type ***
 - This rule type is used to insert comment lines. To designate a line as a comment line, you must enter the rule type *** in column F. Please note that such comment lines cannot be represented or maintained in tree maintenance.
 - Rule type *blank*
 - This rule type defines all other operations.
 - **Variable Key**
 - The variable argument contains field values that you have defined for the decision operations. Other operations can be initiated for such values or value combinations. You can find further information in [Decision Operation/Table Maintenance \[Page 31\]](#).
 - **Status**
 - The status indicates whether a feature has been changed (X) or temporarily stored (S) or whether it must be regenerated (G) if the editor content does not correspond to the generated report or the generated report does not exist.

If you have made changes to a certain feature over a period of time, or if you have copied a standard feature for testing or in order to modify it, you can use the *Compare* function to display these changes.

You can find further information in [Decision Tree of Feature LVNUM: Example \[Page 43\]](#)

Return Operation/Table Maintenance

Return Operation/Table Maintenance

Use

Return operations define the return code of the feature. Return operations are structured as follows:

&<Feature>=<Return code>,



&ABKRS=D5,

Decision Operation/Table Maintenance

Use

To execute a decision operation, select a field from the feature structure.

- **Variant 1:**

The variable key in the follow-on lines contains the field data specified in the decision operation for which you want to specify return codes.

- **Variant 2:**

You do not want to analyze the decision operation fields in their full length. For this reason, you enter the field with offset and length. In this case, the variable key in the follow-on line contains only part of the contents.

- **Variant 3:**

In the decision operation, the field in the feature structure is compared with a value or with another field in the structure. In this case, the variable key in the follow-on line contains a relational operator: > (larger than), < (smaller than), or = (equal to). You can also use * (other) here for all unspecified cases.

A decision operation is structured as follows:

Variable Key	F	D	Operation
		D	Decision operation
Value1			...
Value2			...
...			



Variant 1:

Variable Key	F	D	Operation
		D	BUKRS
1221			&STADT=HEIDELBERG,
2332		D	WERKS
2332 1100			&STADT=BERLIN,
2332 ****			&STADT=FRANKFURT,
****			&STADT=HAMBURG,

Decision Operation/Table Maintenance

The rule is read as follows: If the content of field BUKRS is 1221, then the return code for the feature is HEIDELBERG. If the content is 2332, then check the content of the WERKS field: If the content of the WERKS fields is 1100, then the return code is BERLIN. If the BUKRS field does not contain either 1221 or 2332 (= other), the return code is HAMBURG. If the content of field BUKRS is 2332 but WERKS is not 1100, then the return code is FRANKFURT.

Variant 2:

Variable Key	F	D	Operation
		D	BUKRS +1(2)
22			&STADT=HEIDELBERG,
33		D	WERKS
33 1100			&STADT=BERLIN,
33 *****			&STADT=FRANKFURT,
**			&STADT=HAMBURG,

The rule is read as follows: If the second and third digit of the content of field BUKRS are 22 (offset 1 in length 2), then the return code for the feature is HEIDELBERG. If the content is 33, then check the content of the WERKS field: If the content of the WERKS fields is 1100, then the return code is BERLIN. If this is not 1100, then the return code is FRANKFURT. If the BUKRS field does not contain either 22 or 33, the return code is HAMBURG.

Variant 3:

Variable Key	F	D	Operation
		D	BUKRS ? '1221'
=			&STADT=HEIDELBERG,
>		D	WERKS
> 1100			&STADT=BERLIN,

The rule is read as follows: Compare the contents of the BUKRS field with 1221. If the result is that the company code is 1221, then the return code for the feature is HEIDELBERG. If the result is that the company code is greater than 1221, then check the content of the WERKS field: If the content of the WERKS fields is 1100, then the return code is BERLIN.

Follow-On Operation/Table Maintenance

Use

If one line is not enough for your operations, you can store further operations in the next line using operation NEXTR. The next line must use the same variable key and contain a following-line indicator in column F. Both numbers and letters can be used for following-line indicators. Operation NEXTR is structured as follows:

- **Variant 1:**

Variable Key	F	D	Operation
...			Operation1, Operation2, NEXTR
...	1		Operation3, Operation4, NEXTR
...	2		Operation5, Operation6,

- **Variant 2:**

Variable Key	F	D	Operation
...			Operation1, Operation2, NEXTR A
...	A		Operation3, Operation4, NEXTR
...	B		Operation5, Operation6,

You can use the following symbols as following-line indicators: A, B, C,..., Z, 0, 1, ..., 9. If there is a blank character after operation NEXTR, the number for the next line is determined as follows: B follows A, C follows B,... Line number 1 follows the first blank character. Once line number 9 is reached, no further lines can follow.

Please note that there must be blank characters before NEXTR and between NEXTR and the number/letter.

Subfeature Operation/Table Maintenance

Subfeature Operation/Table Maintenance

Use

You can call up a subfeature with the subfeature operation FLDID. Processing is continued in this subfeature. It is advisable to use operation FLDID when several features can use the same decision tree. This is then created as a subfeature. The subfeature operation is structured as follows:

FLDID=<subfeature>



FLDID=VAC05

Program Operation/Table Maintenance

Use

The program operation offers you the possibility of not explicitly entering the return code of a feature in the decision tree. You can let the ABAP program define the return code.

The program operation is structured as follows:

&<Merkmal>(<Programmname>),



&ABKRS(ZPUZMK00),

You will find the criteria that the ABAP program must fulfill in [Additional Information on Program Operation](#)

[Page 38]

Debugging Operation/Table Maintenance

Debugging Operation/Table Maintenance

Use

You use the BREAK operation to access the user-dependent debugging mode for troubleshooting purposes.

This operation is structured as follows:

BREAK=<username>



```
BREAK=ZAPPA
```

The *user name* must be a valid user name.



Ensure that the debugging operation is always linked to the next line by means of a follow-on operation. This is necessary because a debugging operation is always entered additionally to a decision- or return operation.

Error Operation/Table Maintenance

Use

Operation ERROR does not transmit any back value. Instead, error status 2 is set. Error status 2 displays a constellation that is not permitted in the decision tree.



Variable Key	F	D	Operation
		D	BUKRS
1221			&STADT=HEIDELBERG,
2332			ERROR

Additional Information on Program Operations

Additional Information on Program Operations

A decision tree is usually structured by the decision operations in such a detailed way that a certain return value can be obtained at the end of each decision chain.

The decision tree can become so large that it is no longer possible to obtain an overview of it and maintain it.

Sometimes, reading more information from diverse tables can simplify the specification of the return value.

When these factors are present, you have the possibility of determining the return code in an ABAP program. You transfer the valid fields and contents for the feature to the program.

You must enter the program operation in the table maintenance as follows:

&<feature>(<program>),



&ABKRS(ZPUZMK00),

Create a program operation node in tree maintenance.

A program that is to be called up from a feature must contain a subroutine that defines the return code. Names and parameters differ according to whether the feature provides a return value or return table.

With field return, the subroutine must be named EXT_CALL_F; with table return, the subroutine must be called EXT_CALL_T.

Program structure 1 field return:

REPORT report name.

...

FORM EXT_CALL_F USING NAMEN BACK STATUS *struc* STRUCTURE *struc*.

...

ENDFORM.

Program structure 2 table return:

REPORT report name.

...

FORM EXT_CALL_T TABLES BACK_TAB

USING NAMEN STATUS *struc* STRUCTURE *struc*.

...

ENDFORM.

The parameter NAMEN contains the names of the features. If the same routine is called up by several features, you can use the parameter for case distinction.

The return code should be entered in the parameter BACK during the routine.

The desired return values should be entered in table parameter BACK_TAB during the routine.

You can set the parameter STATUS.

Allowed values:

- STATUS = 0.

Return parameters BACK or BACK_TAB are set. This is the default value.

STATUS = 2

This means that instead of return codes an error should be generated. This is identical to the ERROR operation in the decision tree.

The parameter *struc* must be identical to the feature structure, both in length and structure.



Field transfer:

```
FORM EXT_CALL_F USING NAMEN BACK STATUS PME01 STRUCTURE
PME01.
```

```
IF PME01-WERKS EQ '0001'.
```

```
    BACK = 'ABCD'
```

```
ELSE.
```

```
    BACK = 'EFGH'.
```

```
ENDIF.
```

```
ENDFORM.
```

Table transfer:

```
FORM EXT_CALL_T TABLES BACK_TAB
```

```
        USING NAMEN STATUS PME01 STRUCTURE PME01.
```

```
IF PME01-WERKS EQ '0001'.
```

Additional Information on Program Operations

```
STATUS = 2
ELSE.
BACK_TAB = 'ABCD'.
APPEND BACK_TAB.
BACK_TAB = 'EFGH'.
APPEND BACK_TAB.
ENDIF.
ENDFORM.
```

Features

The program operation can be combined with all other operations in the decision tree.



	D	BUKRS
0001		&ABKRS=D5,
****		&ABKRS(ZPUZMK00),

Editor Commands

Use

A number of commands are available when editing the decision tree for a feature, enabling you to maintain the decision tree and access the functions available in the Editor. You can either enter these commands in the *Command Line* or can access them using the *Feature* and *Edit* menu options.

Features

Command	Meaning
A	(A)ttach: Used as a "goto function". The cursor is moved to the selected line number. Enter "A" followed by a blank character and then the line number you want to access.
CHECK	The syntax of the feature is checked. If it is correct, the system displays a message "Feature OK", and if not, it displays an error message.
F	(F)etch: Enables you to display/maintain a different feature. Enter "F", then a space, and then the five character key of the desired feature.
FIND	Finds the first occurrence of a character string in the feature. Enter "FIND", then a space, and then the character string you are looking for.
I	(I)nsert: Inserts the specified number of blank lines at the end of the Editor. Enter "I", then a space, and then the desired number.
PRINT	Accesses the <i>Print parameter</i> screen. Here, you can choose the output device and the print options.
R	(R)eplace: Replaces one character string with another. The character string is replaced every time it occurs. Enter "R", then a space, then the character string to be replaced, then another space, and finally the character string to be inserted instead.
U	(U)pdate: Saves the feature and checks the syntax at the same time. If an error is identified, you can cancel the update.

Line Commands

Line Commands

Use

Line commands are used to structure and change the source text of decision trees. These commands are entered directly in the *Line number* field of each line of the source text.

Features

Command	Meaning
*	The selected line is displayed as the first line in the Editor.
A	Target line for copy (C/CC..CC) or move (M/MM..MM) command. The lines that have been copied or moved are inserted directly after the selected target line.
B	Target line for copy (C/CC..CC) or move (M/MM..MM) command. The lines copied or moved are inserted directly before the selected target line.
C	Copies the selected line to the target line specified by the command A or B.
CC..CC	Copies the selected line block or line range to the target line specified by the command A or B. Enter CC in the first and last line of the block you want to select.
D	Deletes the selected line.
DD..DD	Deletes the selected line block or line range. Enter DD in the first and last line of the block you want to delete.
I	Inserts a blank line directly beneath the selected line.
In	Inserts n blank lines directly beneath the selected line.
M	Moves the selected lines to the target line specified by the command A or B.
MM..MM	Moves the selected block of lines. Enter MM in the first and last line of the block you want to move.
R	Duplicates the selected line. The copy is inserted in the line directly beneath the original.
Rn	Copies the selected line n times. The copies are inserted directly beneath the original.
X	Copies the content of the X buffer beneath this line.
XX...XX	Writes the selected line block to the X buffer. Enter XX in the first and last line of the block that you want to write to the buffer.

The Decision Tree of Feature LVNUM Example

The Decision Tree of Feature LVNUM Example

Feature LVNUM determines leave entitlement for each leave type. To do this, it refers to variables such as transaction class, country grouping, employee group, and employee subgroup. The structure of this decision tree is very complex and serves as a good example of the use of a subfeature.

	Variable key	F	D	Operations
000010			D	MOLGA
000020	**			FLDID=VAC01
000030	02			FLDID=VAC02 "SWITZERLAND
000040	03			FLDID=VAC03 "AUSTRIA
000050	05			FLDID=VAC05 "NL
000060	08			FLDID=VAC08
000070	09			FLDID=VAC09
000080	12			FLDID=VAC12
000090	22			FLDID=VAC22 "JAPAN

The rule type *D*, which is assigned to field MOLGA in line 000010, means that a decision operation is activated. The leave type and relevant leave entitlement are determined by the country modifier in different subfeatures, which you can call up using operation FLDID.

Line 000020 means: Subfeature VAC01 is called up for all country modifiers that are not explicitly dealt with in this decision tree (** = other).

Continuing from the above example, the source text of subfeature VAC05 is as follows:

	Variable key	F	D	Operations
000010			D	PERSG
000020	*			&LV TYP= , &LV NUM= ,
000030	1		D	PERSK
000040	1 **		D	AGEJJ?'40'
000050	1 ** *			&LV TYP=01/50, &LV NUM=025.00/012.00
000060	1 ** >			&LV TYP=01/50/17, &LV NUM=025.00/012.00/003.00
000070	1 N4			&LV TYP=01, &LV NUM=012.00,

Subfeature VAC05 reads as follows: The contents of field PERSG (Employee group) field must be analyzed. The leave type (LV TYP) and leave entitlement are left *blank* for all unspecified employee groups. If PERSG = 1, the content of the PERSK field (Employee subgroup) must be analyzed. The result of the comparison "Age (field name AGEJJ) compared with 40" must be analyzed for all unspecified employee subgroups. Leave entitlement (LV TYP) 01 and Leave entitlement (LV NUM) 012.00 apply to employee subgroup N4.

On analysis, lines 000040 to 000060 mean the following: If an employee is 40 years of age or younger (*other), his or her leave entitlement is 25 days for leave type 01 and 12 days for leave

The Decision Tree of Feature LVNUM Example

type 50. If an employee is over 40 years of age (shown by the ">" sign in line 000060), he or she gets additional leave type 17 with leave entitlement of 3 days.

Line 000070 means the following (if you include information from the main feature LVNUM): If an employee in The Netherlands (country modifier 05) is assigned to employee group 1 and employee subgroup N4, his or her leave entitlement for leave type 01 is 12 days.

Technical Information: Reading Features in Program

When a feature is activated, a program is generated from the [decision tree \[Ext.\]](#) for the feature. This program contains a subprogram with the following interfaces:

- Feature with field transfer

```
FORM CALL_549B USING BACK STATUS STRUC STRUCTURE struc.
```

...

```
ENDFORM.
```

- Feature with table transfer

```
FORM CALL_549B TABLES BACK USING STATUS STRUC STRUCTURE struc.
```

...

```
ENDFORM.
```

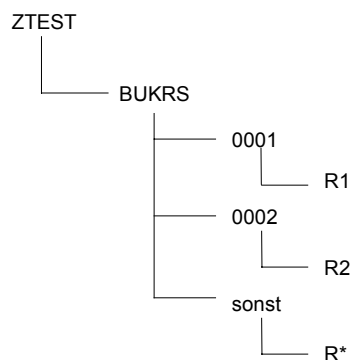


struc is the feature structure. The name of the program is `/1PAPA/FEATcccffff`, where *ccc* is the client and *ffff* is the feature. For example, `/1PAPA/FEAT050ABKRS` is the name of the program that is generated in client `050` when the *Default value for payroll area* feature (ABKRS) is activated, if the feature was previously changed in client `050`.

If the decision tree for a feature contains decision operations, these are listed within the subprogram in CASE statements.



Decision tree of feature `ZTEST` with structure `PME03`.



Technical Information: Reading Features in Program

- Generated program in client 000.

```
PROGRAM /1PAPA/FEAT000ZTEST.  
FORM CALL_549B USING BACK STATUS STRUC STRUCTURE PME03.  
CASE STRUC-BUKRS.  
  WHEN '0001'.  
    BACK = 'R1'.  
  WHEN '0002'.  
    BACK = 'R2'.  
  WHEN OTHERS.  
    BACK = 'R*'.  
ENDCASE.  
ENDFORM.
```

If you want to read the return value for a feature in a program, you have the following options:

- If it is a feature with field return, call the function module HR_FEATURE_BACKFIELD.
- If it is a feature with table return, call the function module HR_FEATURE_BACKTABLE.

For information on the parameters for these modules and how to work with modules, see the documentation for the relevant function modules.



Up to Release 4.0, features can only be read with the include program RPUMKC00 (Call Features and Return Their Values). From Release 4.5A, use the modules HR_FEATURE_BACKFIELD or HR_FEATURE_BACKTABLE.

Maintaining Features

Use

You can maintain features as you wish. Different functions are available to help you do this:

- Create

If an existing standard feature or one which you have modified yourself does not meet your requirements, you can create a new feature. To do this, you must specify the following (in the order listed):

- [Persons Responsible for a Feature \[Page 10\]](#)
- [Country/Component Assignment for a Feature \[Page 11\]](#)
- [Feature Structure \[Page 12\]](#)
- [Decision Tree for a Feature \[Page 16\]](#)
- [Feature Documentation \[Page 15\]](#)

- Change

You can change the individual elements of a feature if necessary. If the source text of a decision tree is changed, you must always check that it is correct.



Changing the structure, the valid fields for a decision operation, or the transfer type can result in changes to the program coding that read the features. You have to reactivate a feature in order for the feature modifications to be effective.

- Copy

The quickest way to create a new feature is to copy an existing feature. If you do this, all elements and control data belonging to the feature are also copied.

- Activate

A feature must be activated before it can transfer a return code (for example, a default value).

- Delete

If you no longer require an existing feature, you can delete it. When doing this, please take care not to delete a feature that is still in use.



Please note that it is not possible to delete individual elements of a feature separately. You can delete the entire feature only.

See also:

[Creating a Feature \[Page 49\]](#)

[Changing a Feature \[Page 60\]](#)

Maintaining Features

[Copying a Feature \[Page 68\]](#)

[Activate a Feature \[Page 71\]](#)

[Deleting a Feature \[Page 75\]](#)

Creating a Feature

Use

You can create a new feature with this function.

Features

Creating a feature is complex process, consisting of five steps.

You create a new feature by creating the following elements:


1. [Create Persons Responsible \[Page 50\]](#)
2. [Creating the Country/Component Assignment for a Feature \[Page 51\]](#)
3. [Creating a Structure \[Page 52\]](#)
4. [Creating the Decision Tree for a Feature \[Page 55\]](#)
5. [Create Feature Documentation \[Page 54\]](#)




The first three steps are absolutely necessary.

Creating the Persons Responsible for a Feature

Creating the Persons Responsible for a Feature

1. Access feature maintenance.
2. Enter the five character key for the feature to which the persons responsible are to be assigned.
3. Under *Subobjects*, flag the *Attributes* field.
4. Choose .

The system displays the *Edit Feature mmmmm: Attributes* screen, where *mmmmm* is the selected feature.

5. In the *Title* field, enter a short description of the feature.
6. Under *Feature*, specify the user name of the person responsible for the feature and indicate whether this person is the only one authorized to make changes to the feature.
7. Under *Structure/field names*, specify the user name of the person responsible for the structure and field names of the feature and indicate whether this person is the only one authorized to make changes to these areas.
8. To save your entries, choose .



If you are already in subobject maintenance (such as a decision tree, structure, country - component assignment, or documentation), you can jump to the maintenance screen for responsible persons by choosing *Goto* → *Persons responsible*.

Result


You have assigned the persons responsible for the feature. The date of the entries appears in the field *Created on*. This date changes when new entries are made in this screen.

Creating the Country/Component Assignment for a Feature

Prerequisites

Creating the country/component assignment is the second step in creating a feature. You cannot maintain the country/component assignment until you have created the persons responsible for the feature.

Procedure

1. Access feature maintenance.
2. Enter the five-character key of the feature for which you want to create a country/component assignment.
3. Under *Subobjects*, flag the *Attributes* field.
4. Choose  *Change*.



The system displays the *Edit Feature xxxxx: Attributes* screen, where xxxxx is the selected feature.


5. Choose Country/Component Assignment.

The system displays a screen entitled *Edit Feature xxxxx :Country/Component*

6. Under *Country Assignment*, select all countries in which the feature is to be used.
7. Under *Component Assignment*, select all fields in the HR component to which you want the feature to apply.



You can scroll through the entries in both sections using the *Previous*  / *Next page*  functions.

8. To save your entries, choose .

Result

You have created the country/component assignment of the feature. You can now create the structure and decision tree of the feature.

Creating a Structure


Creating a Structure

Prerequisites


Creating the structure is the third necessary step in creating a feature. You must first carry out the following steps:

- [Create Persons Responsible \[Page 50\]](#)
- [Creating the Country/Component Assignment for a Feature \[Page 51\]](#)

Procedure

1. Access feature maintenance.
2. Enter the five-character key of the feature whose structure you want to create.
3. Under *Subobjects*, flag the *Attributes* field.
4. Choose  *Change*.

The system displays the *Edit Feature xxxxx: Attributes* screen, where xxxxx is the selected feature.
5. Choose *Structure*.

The system displays a screen entitled *Edit Feature xxxxx : Structure*, where xxxxx is the selected feature.
6. Enter the five-character key of the structure containing the fields required for the feature.
7. Enter the structure name for the return code, or name a field for the return code. This step is optional.
8. Set the transfer type.
9. Specify whether the feature is to be used as a subfeature (X = yes, SPACE = no).
10. Under the header *Valid field names for decisions*, flag the relevant field names for the feature.
11. To save your entries, choose .

Result

You have created the feature structure, and can now [create the decision tree for the feature \[Page 55\]](#), as well as [create documentation for the feature \[Page 54\]](#), which is an optional step.

See also:

[Creating the Structure for a New Feature \[Page 53\]](#)



Creating a Structure for a New Feature

Prerequisites

You want to create a feature entitled TEST2 that returns a default value. You have already maintained all elements apart from the structure. Now you must create a structure for the feature, so that you can create a decision tree. Please note the following criteria:

- The return code will be a field transfer (transfer type 1)
- The feature will be used as a main feature
- Structure PME04 contains the fields that you require for decision operations
- The default value will be based on company code, employee group, employee subgroup, country grouping, and transaction class
- The return code will be a payroll area from the Payroll Area table (T549A)

Procedure

1. Access feature maintenance.
2. Enter the five-character key (TEST2) for the feature whose structure you wish to create.
3. Under *Subobjects*, flag the *Attributes* field.
4. Choose  *Change*.
The system displays the *Edit Feature TEST2 : Attributes* screen.
5. Choose *Structure*.
The system displays a screen entitled *Edit Feature TEST2 : Structure*.
6. Enter the five-character key (PME04) for the structure that contains the fields required for the feature.
7. Define the *Transfer type* as 1 for *field transfer*.
8. Do not set the *Subfeature* indicator.
9. Enter P0001-ABKRS in the field *Field name for return code*.
10. Under the header *Valid field names for decisions*, flag the relevant field names for the feature.
11. To save your entries, choose .



Result

You have created the feature structure. Now you can create a decision tree for the feature.

Creating Documentation


Creating Documentation

Prerequisites


The following procedure only applies if you have created a new schema using , not , since this means there is no existing documentation. It is also assumed here that you have already assigned and saved persons responsible for the feature [Create Persons Responsible \[Page 50\]](#)

Procedure


Access feature maintenance.

1. Enter the five-character key of the feature for which you want to create documentation.
2. Under *Options*, flag the *Documentation* field.
3. Choose  *Change*.

The *Change HR System Documentation <title>* screen appears. You are now in the Editor.

4. Enter your documentation.
5. To save your entries, choose .



Once the persons responsible have been assigned and the assignment saved, a feature is marked as created. For this reason, the  *Change* function is used to create further elements for the feature.

Result

You have created the feature documentation.

Creating the Decision Tree for a Feature

Prerequisites

Creating a decision tree is the penultimate step in creating a new feature.

You cannot create a decision tree until you have performed the following steps:

- [Create Persons Responsible \[Page 50\]](#)
- [Creating the Country/Component Assignment for a Feature \[Page 51\]](#)
- [Creating a Structure \[Page 52\]](#)

You can create a decision tree in either tree maintenance or table maintenance.



SAP recommends using tree maintenance.

Procedure

To create the decision tree for a feature, see the following sections:

[Creating the Decision Tree of a Feature \(Tree Maintenance\) \[Page 56\]](#)

[Creating the Decision Tree of a Feature \(Table Maintenance\) \[Page 58\]](#)

Creating the Decision Tree for a Feature (Tree Maintenance)


Creating the Decision Tree for a Feature (Tree Maintenance)

Prerequisites

Creating a decision tree is the penultimate step in creating a new feature. You create the decision tree after you have performed the following steps:

- [Create Persons Responsible \[Page 50\]](#)
- [Creating the Country/Component Assignment for a Feature \[Page 51\]](#)
- [Creating a Structure \[Page 52\]](#)


Procedure

1. Access feature maintenance.
2. Enter the five character key of the feature whose decision tree you want to create.
3. Under *Subobjects*, flag the *Decision tree* field.
4. Choose  *Change*.

You access the *Edit Feature xxxxx: Decision Tree* screen, whereby xxxxx is the selected feature.






In the menu, you can change the maintenance mode by choosing *View* → *Tree maintenance* or *View* → *Table maintenance*.

5. Place the cursor on the feature name, and choose . Select a suitable node type, and let yourself be guided by the system. Edit the decision tree with the help of the tree functions until it fulfills your requirements.



You can find further information on nodes in [Decision Tree in Tree Maintenance \[Page 17\]](#).

6. Check whether the feature is syntactically correct by choosing .
7. Make the necessary changes.
8. To save the feature, choose .
9. To activate the feature, choose .

See also:

[Activate a Feature \[Page 69\]](#)

Result

You have created a decision tree in tree maintenance. You can now [create the documentation for the feature \[Page 54\]](#) (this is an optional step).

Creating the Decision Tree for a Feature (Tree Maintenance)

Creating the Decision Tree for a Feature (Table Maintenance)


Creating the Decision Tree for a Feature (Table Maintenance)

Prerequisites

Creating a decision tree is the penultimate step in creating a new feature. You create the decision tree after you have performed the following steps:

- [Create Persons Responsible \[Page 50\]](#)
- [Creating the Country/Component Assignment for a Feature \[Page 51\]](#)
- [Creating a Structure \[Page 52\]](#)




Procedure

1. Access feature maintenance.
2. Enter the five character key of the feature whose decision tree you want to create.
3. Under *Subobjects*, flag the *Decision tree* field.
4. Choose  *Change*.

You access the *Edit Feature xxxxx: Decision Tree* screen whereby xxxxx is the selected feature. The lines in the source text are blank.



In the menu, you can change the maintenance mode by choosing *View → Tree maintenance* or *View → Table maintenance*.

5. Create the decision tree by entering suitable operations, variable keys, and field names.
The [Editor Commands \[Page 41\]](#) and [Line Commands \[Page 42\]](#) in the Editor for feature decision trees can be of some assistance in this procedure.
6. Check whether the syntax of the source text for the decision tree is correct. Choose  or, alternatively, enter the command CHECK in the command line.
7. Make the necessary changes.
8. To save your entries, choose .
9. To activate the feature, choose .

See also:

[Activate a Feature \[Page 69\]](#)

Result

You have created a decision tree in table maintenance. You can now [create the documentation for the feature \[Page 54\]](#) (this is an optional step).

Creating the Decision Tree for a Feature (Table Maintenance)

Changing a Feature

Changing a Feature

Use

Changing a feature involves changing its individual elements.



Check whether the changes in the decision tree are syntactically correct for the feature.

See also:

[Changing the Persons Responsible for a Feature \[Page 61\]](#)

[Changing the Country/Component Assignment for a Feature \[Page 62\]](#)

[Changing the Structure of a Feature \[Page 63\]](#)

[Changing the Decision Tree of a Feature \[Page 65\]](#)

[Changing the Feature Documentation \[Page 64\]](#)


Changing the Persons Responsible for a Feature

Prerequisites


To change data on the *Person Responsible/Status* screen, you must be the person responsible assigned to this feature. Exception: If the *Changes only by person responsible* field is not flagged for this feature.

You also have the possibility of changing the name that you have assigned to the feature.

Procedure

1. Access feature maintenance.
2. Enter the five character key of the feature whose responsible persons you want to change.
3. Under *Options*, flag the *Persons responsible/status* field.
4. Choose  *Change*.

You access the *Edit Feature xxxxx: Person Responsible/Status* screen whereby xxxxx is the selected feature.


5. If applicable, change the name of the feature in the field *Title*.
6. Make the desired changes in the *Feature* and *Structure/field* names areas.
7. To save your entries, choose .

Result

You have changed the assignment of persons responsible. The date and time of the last change are now those of your last save. The version number increases by one.

Changing the Country/Component Assignment for a Feature

Changing the Country/Component Assignment for a Feature

1. Access feature maintenance.
2. Enter the five-character key for the feature whose country/component assignment you want to change.
3. Under *Subobjects*, flag the *Attributes* field.
4. Choose  *Change*.



The system displays the *Edit Feature xxxxx : Attributes* screen, where xxxxx is the selected feature.


5. Choose *Country/Component Assignment*.

The system displays a screen entitled *Edit Feature xxxxx : Country/Component*

6. Make the necessary changes to the selected entries under *Country assignment* and *Component assignment*.



You can scroll through the entries in both sections using the *Previous*  / *Next page*  functions.

7. To save your entries, choose .

Result


You have changed the country/component assignment of the feature.

Changing the Structure of a Feature

Prerequisites

Only the person responsible for the feature should make changes to the structure screen. Note that every change made to the structure affects the programs that read the feature.


Procedure

1. Access feature maintenance.
2. Enter the five-character key of the feature whose structure you want to change.
3. Under *Subobjects*, flag the *Attributes* field.
4. Choose  *Change*.

The system displays the *Edit Feature xxxxx: Attributes* screen, where xxxxx is the selected feature.

5. Choose *Structure*.

The system displays a screen entitled *Edit Feature xxxxx : Structure*.

6. Make the necessary changes.
7. To save your entries, choose .

Result


You have changed the structure of the feature.




Every change to the structure of a feature affects the decision tree. Check that the decision tree is syntactically correct.

Changing Documentation

Changing Documentation

1. Access feature maintenance.
2. Enter the five character key of the feature whose documentation you want to change.
3. Under *Options*, flag the *Documentation* field.
4. Choose  *Change*.


The system displays the *Change HR System Documentation <title>* screen.

5. Make the required changes to the documentation.
6. To save your entries, choose .

Result

You have changed the feature documentation.




Changing the Decision Tree of a Feature

1. Access feature maintenance.
2. Enter the five character key of the feature whose decision tree you want to change.
3. Under *Subobjects*, flag the *Decision tree* field.
4. Choose  *Change*.

You access the *Edit Feature xxxxx: Decision Tree* screen whereby xxxxx is the selected feature.

5. Change the decision tree by changing the appropriate operations, variable keys, and field names.

The [Editor Commands \[Page 41\]](#) and [Line Commands \[Page 42\]](#) in the Editor for feature decision trees can be of some assistance in this procedure.

6. To save your entries, choose .
7. To check whether the syntax of the source text for the decision tree is correct, choose .
8. Make any necessary changes and save your entries again.
9. To activate the feature, choose .

Result

You have changed the decision tree.

Technical Information: Changing a Standard Feature

Technical Information: Changing a Standard Feature

If SAP makes any changes to a [standard feature \[Ext.\]](#), the necessary information can be found in the release notes provided. If the changes are critical changes, SAP also creates a note.

As a rule, any changes made to a decision tree are not critical. However, we will inform you of these changes, so that you can make your own decision as to how you implement the new standard feature.

Any changes made to additionally selected fields in the feature structure are also not critical, but we will also inform you of the changes in the relevant release note.

However, any changes made to the feature structure, or to whether a field or table return is defined for a specific feature, are classed as extremely critical. In this case, we take particular care to ensure that you are informed of these changes. Please make sure to check our notes as well as our release notes.

See also:

[Technical Information: Activating a Feature \[Page 72\]](#)

Copy a Feature

Use


The quickest way to create a new feature is to copy an existing feature. When this is done all elements and control data are also copied.

See also:


[Copying a Feature \[Page 68\]](#)

Copying a Feature

Copying a Feature

1. Access feature maintenance.
2. Enter the five character key of the feature you want to copy.
3. Choose .

The system displays the *Copy* dialog box. The selected feature is displayed in the fields *From feature* and *To feature*.

4. Enter the five character key of the feature you want to create by copying in the *To feature* field.
5. Choose .

The system displays a message *Feature mmmmm has been copied*, where *mmmmm* stands for the original feature selected on the initial screen. All elements of the feature have been copied.

Result

You have copied the feature.

Activate a Feature

Use

A feature must be activate before it can produce a return code (for example, a default value).

Prerequisites

You can only activate a feature if the feature is error-free.

See also:

[Activating a Feature \[Page 71\]](#)

Activate a Feature

Activation Types

Purpose

There are four activation types. You can specify the activation type by stating the feature to be activated.

Complete activation

The originals of the subfeatures are called up and activated. Complete activation then takes place. A complete activation of all features always takes place. You are not permitted to enter a feature for the selection option *Feature(s)*.

Activation of selected features

All selected features are activated whether or not activation is necessary for each of them. It is always the last saved version of the relevant feature that is activated.


Activation of changed features

Only the modified features that are selected are activated. These must be activated before they can be used. If you save or temporarily store these features without checking, this does not count as a change.

Internal activation



Only for internal SAP use.

Activating a Feature

1. Access feature maintenance.
2. Enter the five character key of the feature, and choose .

The activation log is created. It contains a list of the features activated and activation statistics.



If you do not enter the five-character key of the feature before choosing , the system displays the *Generate Features* screen. Enter the name of the features to be activated, and the required selection criteria, and then select . The activation log is created. It contains a list of the features activated and activation statistics.

Result

You have activated the feature.

Technical Information: Activating a Feature

Technical Information: Activating a Feature

When you activate a feature, the system either creates a feature program from the [decision tree \[Ext.\]](#) of the feature, or creates/updates the entry in the *Feature Directory* (T549D).



The system does not create a feature program if the decision tree only contains one return value, which does not exceed a specific length (release-dependent). In this case, this return value is stored in the FUNCT field of the *Feature Directory* (T549D), and the FUNID field is filled with V (value), and not P (program).

You can activate a feature in one of the following ways:

- Activate a feature from the decision tree (tree or table maintenance)

If you choose this method, the system checks the current decision tree, and saves it (updating the S and U entries in the *Company Features* (T549B) or *Decision Trees for Features (Customers)* (T549C) tables), before creating the feature program and the entry in the *Feature Directory* (T549D).
- Activate a feature by running report RPUMKG00

You can activate this feature either within feature maintenance (transaction PE03 - select Activate), or by running report RPUMKG00 (*Generate (Activate) Features*) from the program execution selection screen (transaction SA38 or SE38). The system will then create the feature program from the U version of the *Company Features* (T549B) or *Decision Trees for Features (Customers)* (T549C) tables.
- Activate a customer feature automatically upon deletion (as of release 4.6B)

If you delete a standard feature that you have changed in a non-standard client, the system will delete the corresponding entries from the *Decision Trees for Features (Customers)* table (T549C), before updating the entry in the *Feature Directory* (T549D). In other words, the system takes the name of the feature program or the corresponding return value from the standard client, and enters it for the corresponding entry in the *Feature Directory*.
- Activate a customer feature automatically upon transport (as of release 4.6B)

For standard and customer features, there is a logical transport object with an *After-Import-Method* (PMKS or PMKC). All transported features are automatically activated in the target system. When you transport customer features, the corresponding feature is activated in the target system, but only in the same client from which it was transported. When you transport standard features, they are activated in the standard client of the target system first, before they are activated in all other clients in the system.



Caution: Any changes made to standard features affect the corresponding standard client of the target system. They do not affect clients in which the corresponding feature was modified, as the feature is also activated if it is based upon the corresponding U entries of the *Decision Trees for Features (Customers)* table (T549C), and not of the *Company Features* table (T549B).

If SAP makes any changes to standard features, the necessary information can be found in the release notes provided. You can then decide, for example, if you need to

Technical Information: Activating a Feature

make the changes yourself manually, or whether you need to maintain your decision tree again, using the new standard feature as the basis.

Deletion of a Feature

Deletion of a Feature

If you no longer require an existing feature, you can delete it. When doing this, take care not to delete a feature that is still in use.




Please note that it is not possible to delete individual elements of a feature separately. You can delete the entire feature only.


See also:

[Deleting a Feature \[Page 75\]](#)

Deleting Features

1. Access feature maintenance.
2. Enter the five-character key of the feature you want to delete.
3. Choose .

The system displays the *Delete* dialog box. The feature you selected is displayed in the appropriate field.

4. Choose .

The system displays a message *Feature mmmmm has been deleted*, where *mmmmm* stands for the selected feature.

Result

You have deleted the feature.



Please note that deleting a feature will probably mean making modifications in all the areas in which the feature was used.

Environment for Maintaining Features

Environment for Maintaining Features

Purpose

The following processes support you in feature maintenance:

- Accessing feature maintenance
- Modification concept
- Transport connection

Process flow

- **Accessing feature maintenance**

1. You can access maintenance of the decision tree for a feature directly from the IMG (Implementation Guide) for *Personnel Management*. If you do this, the system always accesses the feature assigned to that particular IMG module. If you want to change the structure, country/component assignment, persons responsible, or the feature documentation, choose *Goto*.
2. You can also call up the feature maintenance transaction, **PE03**, directly. This accesses the initial *Features* screen:
3. To maintain a different feature, choose *Feature* → *Other feature* from the decision tree. However, this function is only possible within table maintenance.

- **Modification concept**

In the standard system, the *Human Resources* component (HR) contains a number of features that you can copy and modify according to your requirements. The modified versions are those used in the customer's live system. This means that there are two versions of the same feature with the same name.

If you call up a feature on the initial screen to display or edit it, the system only accesses the modified version. The standard SAP feature of the same name is not affected by any changes you might make.

The *Compare* function enables you to view the standard feature and identify any differences between it and the modified version.

The standard features may change with each new release of the SAP System. Although this may not always affect your configured system, it is important to examine and take note of such changes, as you may wish to change your modified versions accordingly.

- **Transport connection**

All changes that you make to a feature must be transported from your customizing client into the other clients for your SAP System. Depending on the configuration of your customizing client, all changes can be automatically included in one transport request. If this is not possible, you must create a separate transport request for each feature to which you make changes in order to be able to transport it. To do this, access the decision tree editor (*Edit Feature* screen) and follow the menu path *Feature* → *Transport* → *Include in order*.

Technical Information: Saving a Feature

Saving Standard and Customer Features

Any features created and delivered by SAP ([standard features \[Ext.\]](#)), are created in the standard client (000), and saved in the *Company Features* table (T549B). [Customer Features \[Ext.\]](#) are saved in the *Decision Trees for Features (Customers)* table (T549C).

Saving a Modified Standard Feature

If a customer modifies a standard feature, the system copies the relevant entries from the *Company Features* table (T549B), and uses them as a basis for the changes. The modified feature is then saved in the *Decision Trees for Features (Customers)* table (T549C). This does not affect the standard feature.

Changing the Standard Client

You can change the standard client in the include program **RPDMMND00** (SAP standard client).

Information on Tables T549B and T549C

The NAMEN field contains the feature names.

Meanings of contents of field KENID:

- D: Administration information (such as last change user, feature structure, field or table return, for example). You can find this information in the FUNCO field.
- F: Fields you can use in the decision tree. You can find the corresponding field names in the VARGU field. Fields that have an indicator set in the FUNCO field are used in decision operations within activated features.
- G: This entry only exists if the feature is not currently active. In other words, if the feature has been changed since it was last activated, or if it has never been activated.
- S: These entries contain the last version of the feature (with or without check). If you access the feature tree, this feature is displayed.

For each line in the table editor, there are 2 S lines: SEQNO 1, and SEQNO 2. If the FUNCO field contains SEQNO 1, it contains the data from the Variable Argument field. If the FUNCO field contains SEQNO 2, however, it contains the data from the Operations field.

- U: These entries contain the last version saved after being checked. This version corresponds to the activated feature. If a feature is activated via the *Generate (Activate) Features* report (RPUMKG00), the system activates the U version.

Utilities and Tools

Utilities and Tools

- **The feature directory**

The Feature Directory is a report (**RPUMKD00**). This report generates a list of features according to your specific search criteria. The search criteria that are available for the feature directory include the following:

- Name of the feature
- Persons responsible for feature
- Type of feature
- Version of feature
- Generation indicator for feature

The list generated by the report provides the following information for each entry:

- Name
- Version
- Description
- Persons responsible
- Status of each entry

You can display the attributes and documentation on each of these features in the results list. In addition, you can access the *Initial Screen* of the feature you have flagged. To do this, choose *Edit*. You can then display or change the individual elements of the feature.



When maintaining the search criteria, you can use many of the functions available on the HR report *Request screen*.

- **Compare function**

The compare function enables you to compare the contents of two features. You can either compare two different features or compare the standard version of the feature (from the standard SAP system) with its modified version (used by customer).

For further information on standard or modified features, see [Environment of Feature Maintenance \[Page 76\]](#).

When you call up this function, the system automatically selects the feature that you are editing or displaying as one of the two features to be compared. The second feature can then be entered in a dialog box that displays various processing options.

Once the comparison has taken place, the following is displayed:

- Transfer type for both features
- Authorized fields in both features
- Persons responsible for both features
- Administrative data on both features

- Results of the comparison

These results display such information as changed, inserted, or deleted lines.

When comparing features, the following processing options are available:
- **Use standard feature**

This option enables you to compare a modified feature (the feature in the Editor) with a standard feature. If you do not choose this option, the system uses the modified version.
- **Display differences only**

Choose this option if you only want the system to display the differences that it found after the compare. If you do not choose this function, then both the differences and unchanged sections are displayed.
- **Restore function**

The restore function overwrites the editor content/decision tree with the last saved version of the feature.

See also:

[Creating a Feature Directory \[Page 80\]](#)

[Comparing Features \[Page 81\]](#)

Creating a Feature Directory

Creating a Feature Directory


1. Access feature maintenance.
2. Choose *Feature* → *Overview*.

The system displays the *Feature Directory* screen.

3. Enter your search criteria.



There are a number of functions that will help you in this step. They are the same as those on the HR report *Request Screen*.

4. Choose .

Result




You have created a feature directory.

Comparing Features

Prerequisites

The only place that you can compare features is in table maintenance.

Procedure

1. Access feature maintenance.
 2. Enter the five-character key of the feature you want to include in the compare.
 3. Under *Subobjects*, flag the *Decision tree* field.
 4. Choose  *Display* or  *Change*.
- The system accesses the source text of the feature selected.
5. Select *Feature* → *Compare*.
- The *Compare* dialog box appears. You see the five character key of the feature you specified on the *Initial screen*.
6. If you want to display the differences between this feature and the standard feature, flag the field *Use standard feature*. If you want to display the differences between this feature and the last saved version, flag the field *Display differences only*.
 7. Specify the feature you want to compare with that selected on the *Initial screen*.
 8. Choose .

The system displays the *Compare features* screen. The system displays the following for each of the two features:

- Attributes
 - Transfer type
 - Structure
 - Person responsible for structure
 - Person responsible for feature
- Administrative data
 - Creation date
 - Last changed on
 - Version
- Result of the comparison

The system displays one of the following messages: *The following lines were inserted* or *Contents unchanged*.

Result

You have carried out the comparison.

Comparing Features