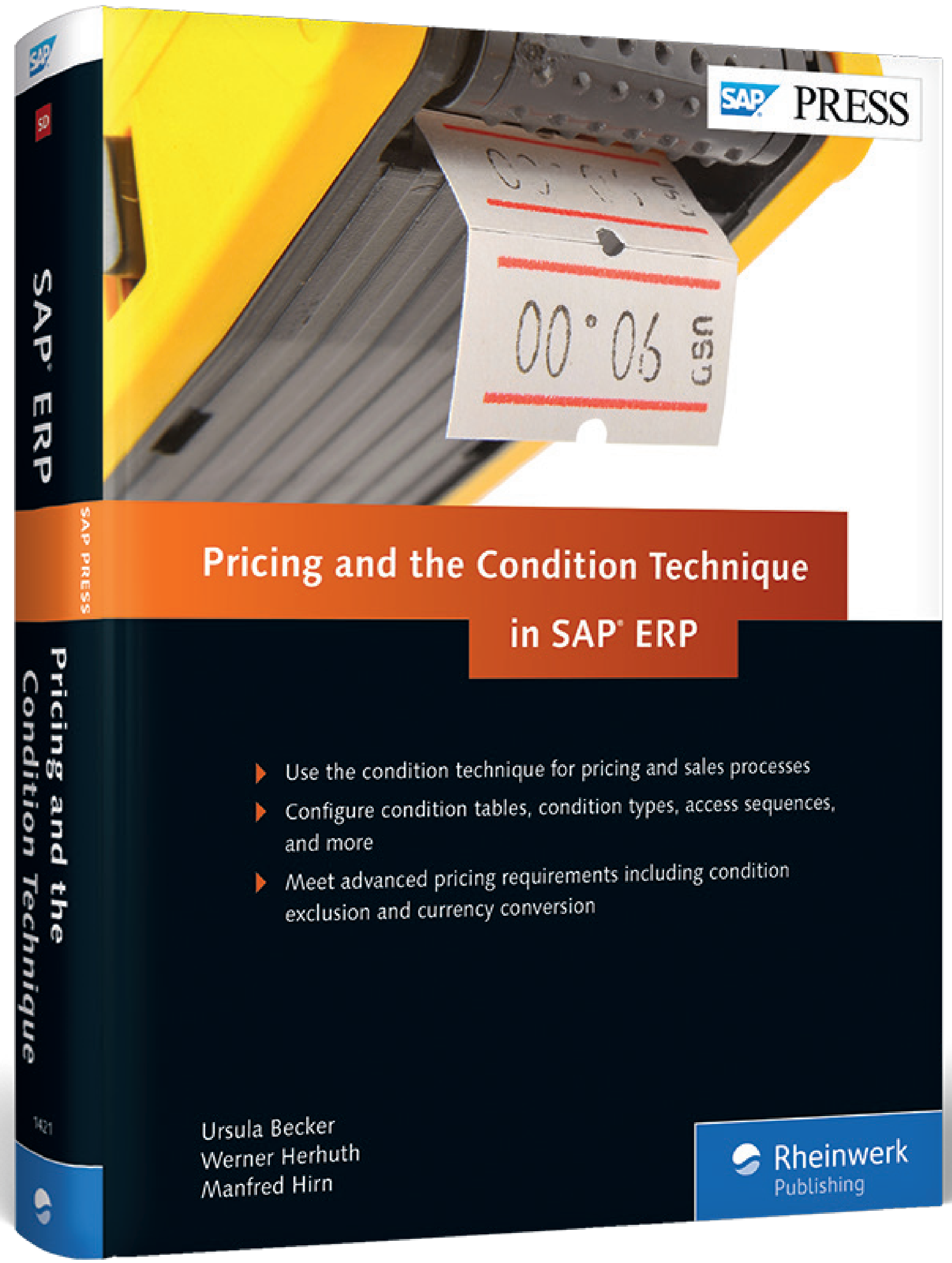
SAP PRESS

First-hand knowledge.

Rheinwerk Publishing


SAP PRESS
SAP ERP
Pricing and the Condition Technique in SAP ERP
Use the condition technique for pricing and sales processes
Configure condition tables, condition types, access sequences, and more
Meet advanced pricing requirements including condition exclusion and currency conversion
Ursula Becker
Werner Herhuth
Manfred Hirn
Rheinwerk Publishing

Reading Sample

This sample chapter discusses executing, configuring, and using pricing reports. It also covers the worklist provided for the internal sales representative, the use of the net price list, and the performance-optimized price list.

"Condition Lists"

Contents

Index

The Authors

Ursula Becker, Werner Herhuth, Manfred Hirn

Pricing and the Condition Technique in SAP ERP

665 Pages, 2017, $89.95
ISBN 978-1-4932-1421-1

www.sap-press.com/4173

Contents

PART I Foundations

PART II Standard Configuration

PART III Advanced Techniques, Tips, and Tricks

11 System Adaptation Using Requirements, Formulas, and User Exits ... 385

12 Typical Practice Demands on Pricing and Their Solutions ... 431

PART IV Rebate Processing in Sales

Like any other master data, condition records must also be maintained by the respective departments of the company. Efficient tools can select the relevant subset from a large population of condition records to assist in this task.

3 Condition Lists

In addition to the different maintenance transactions, additional tools are provided to maintain condition records and to evaluate them in order to answer different questions. For example, you can use pricing reports or personalized worklists to get an overview of condition records or use price lists for price information.

In the following sections, we will take a look at executing, configuring, and using pricing reports. We will also discuss worklist provided for the internal sales representative role (`SAP_BPR_INT_SALES_REP_14` or `SAP_SR_INT_SALES_REP_5`), the use of the net price list, and the performance-optimized price list.

3.1 Pricing Reports

To get an overview of the condition records in the system, you can use special pricing reports. For example, the condition records can be searched to answer the following questions:

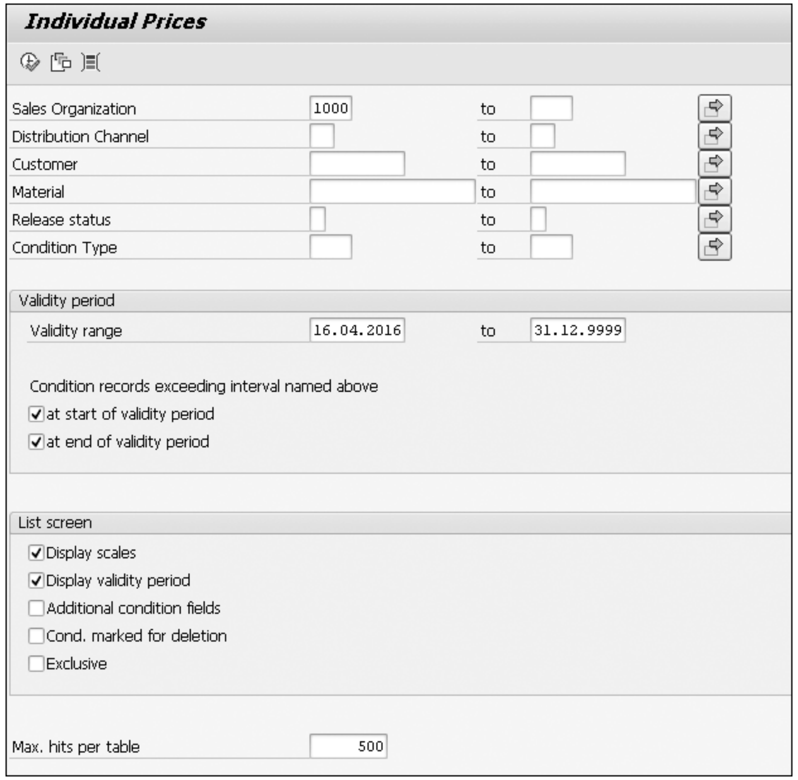
- Which customer-specific price agreements exist in my sales organization for specific products?
- Which condition records are stored in the system for certain *incoterms*?
- What conditions have been created for a specific customer?

In the standard SAP system, there are a number of preconfigured pricing reports, such as pricing report 16 (Individual Prices). In addition, you can configure further pricing reports by a report generator within Customizing (see Section 3.1.2) to cover any customer requirements. Let us now take a look at how to handle pricing reports.

3.1.1 Execution of Pricing Reports

To call a pricing report, choose the menu path SAP Easy Access • Logistics • Sales and Distribution • Master Data • Conditions • List • Pricing Report (Transaction V/LD).

Let's look at this tool using the example of pricing report 16 (see Figure 3.1): Initially, after starting a pricing report, a selection screen appears to restrict the query. The number of selection fields can be configured, as you will also see in Section 3.1.2, when creating the pricing report in Customizing. You can also configure the default for the attributes in the field group List screen.



Individual Prices

Sales Organization	1000	to	
Distribution Channel		to	
Customer		to	
Material		to	
Release status		to	
Condition Type		to	

Validity period

Validity range	16.04.2016	to	31.12.9999

Condition records exceeding interval named above

☑ at start of validity period

☑ at end of validity period

List screen

☑ Display scales

☑ Display validity period

☐ Additional condition fields

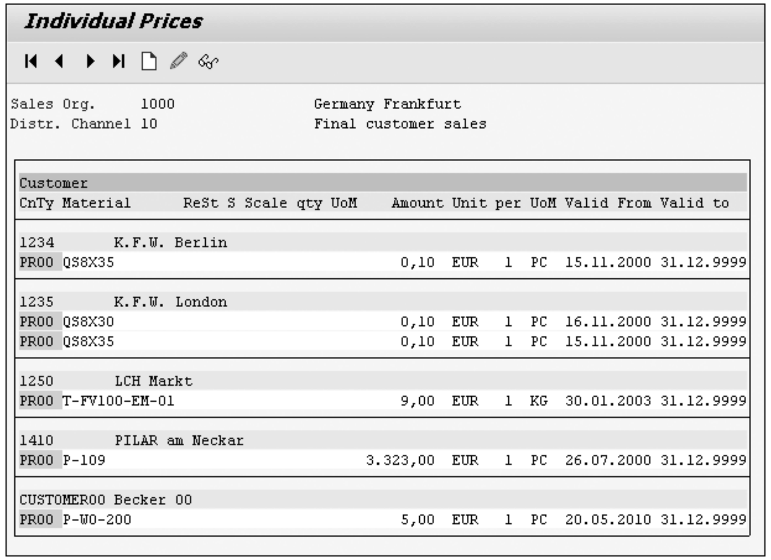
☐ Cond. marked for deletion

☐ Exclusive

Max. hits per table 500

Figure 3.1 Calling the Individual Prices Pricing Report (Transaction V/LD)

After entering the selection criteria, you can start the pricing report by selecting Execute or using F8. The results will displayed (see Figure 3.2). Within the possibilities offered, the appearance of the result list (e.g., positioning a field on the page header, group header, or item level) can be configured in the Customizing of the pricing report.



Individual Prices

```
Sales Org.     1000              Germany Frankfurt
Distr. Channel 10                Final customer sales

Customer
CnTy Material      ReSt S Scale qty UoM    Amount Unit per UoM Valid From Valid to

1234       K.F.W. Berlin
PR00 QS8X35                                  0,10  EUR   1  PC  15.11.2000 31.12.9999

1235       K.F.W. London
PR00 QS8X30                                  0,10  EUR   1  PC  16.11.2000 31.12.9999
PR00 QS8X35                                  0,10  EUR   1  PC  15.11.2000 31.12.9999

1250       LCH Markt
PR00 T-FV100-EM-01                           9,00  EUR   1  KG  30.01.2003 31.12.9999

1410       PILAR am Neckar
PR00 P-109                               3.323,00  EUR   1  PC  26.07.2000 31.12.9999

CUSTOMER00 Becker 00
PR00 P-W0-200                                5,00  EUR   1  PC  20.05.2010 31.12.9999
```

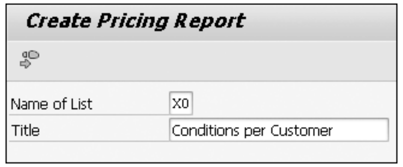
Figure 3.2 Individual Prices: Result List

From the results list, you can branch to the condition maintenance Transactions VK11 (Create), VK12 (Change), and VK13 (Display). In the case of display and change, you jump directly to the condition record marked in the list.

3.1.2 Configuration of a Pricing Report

The configuration of pricing reports cannot be done in the application menu of the end user, but it is part of Customizing. The settings for the pricing reports can be accessed via the menu path IMG • Sales and Distribution • Basic Functions • Pricing • Maintain Pricing Report.

To create a new pricing report, you first select a two-digit abbreviation from the customer namespace (these are, by definition, all two-digit combinations that begin with a letter) and a title. In our example report, we will choose the abbreviation X0 and the title Conditions per Customer (see Figure 3.3).

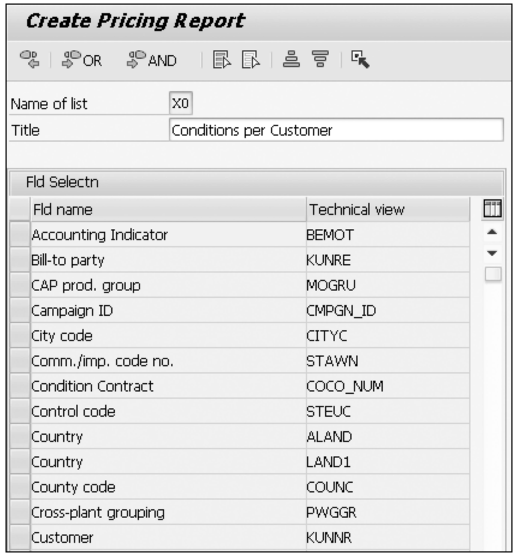


Create Pricing Report

Name of List	X0
Title	Conditions per Customer

Figure 3.3 Create Pricing Report: Initial Screen

Next, a screen appears with all fields that are part of a condition table in the underlying development system (see Figure 3.4).



Create Pricing Report

Name of list	X0
Title	Conditions per Customer

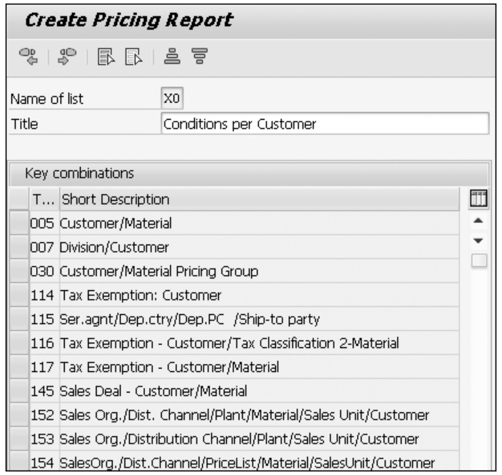
Fld Selectn

Fld name	Technical view
Accounting Indicator	BEMOT
Bill-to party	KUNRE
CAP prod. group	MOGRU
Campaign ID	CMPGN_ID
City code	CITYC
Comm./imp. code no.	STAWN
Condition Contract	COCO_NUM
Control code	STEUC
Country	ALAND
Country	LAND1
County code	COUNC
Cross-plant grouping	PWGGR
Customer	KUNNR

Figure 3.4 Create Pricing Report: Field Selection

When selecting the fields, the challenge is to make a preliminary decision: Which condition tables and condition types are to be evaluated in the resulting pricing report? Here, a combination of the fields with AND or with OR is possible. Each option has an impact on the selection of the condition tables. We opt for the Customer (KUNNR) field. As a result, we receive a listing of all condition tables in which field KUNNR is included, as shown in Figure 3.5.

We select condition tables `005`, `007`, `305`, and `307` and come to the last screen in the generation sequence (see Figure 3.6), where we can influence the selection screen and the appearance of the result list.



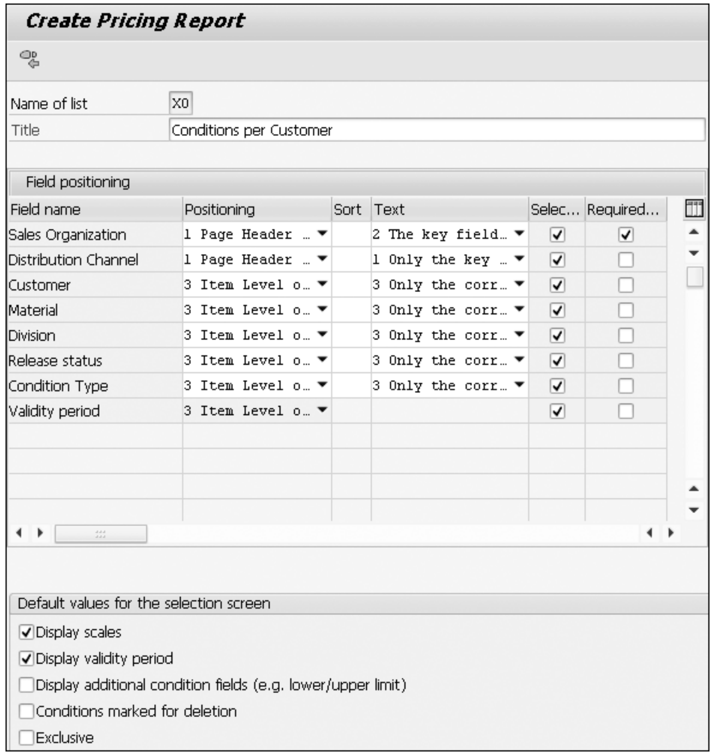
Create Pricing Report

Name of list	X0
Title	Conditions per Customer

Key combinations

T...	Short Description
005	Customer/Material
007	Division/Customer
030	Customer/Material Pricing Group
114	Tax Exemption: Customer
115	Ser.agnt/Dep.ctry/Dep.PC /Ship-to party
116	Tax Exemption - Customer/Tax Classification 2-Material
117	Tax Exemption - Customer/Material
145	Sales Deal - Customer/Material
152	Sales Org./Dist. Channel/Plant/Material/Sales Unit/Customer
153	Sales Org./Distribution Channel/Plant/Sales Unit/Customer
154	SalesOrg./Dist.Channel/PriceList/Material/SalesUnit/Customer

Figure 3.5 Create Pricing Report: Selection of Condition Tables



Create Pricing Report

Name of list	X0
Title	Conditions per Customer

Field positioning

Field name	Positioning	Sort	Text	Selec...	Required...
Sales Organization	1 Page Header ...		2 The key field...	☑	☑
Distribution Channel	1 Page Header ...		1 Only the key ...	☑	□
Customer	3 Item Level o...		3 Only the corr...	☑	□
Material	3 Item Level o...		3 Only the corr...	☑	□
Division	3 Item Level o...		3 Only the corr...	☑	□
Release status	3 Item Level o...		3 Only the corr...	☑	□
Condition Type	3 Item Level o...		3 Only the corr...	☑	□
Validity period	3 Item Level o...			☑	□

Default values for the selection screen

☑ Display scales
☑ Display validity period
□ Display additional condition fields (e.g. lower/upper limit)
□ Conditions marked for deletion
□ Exclusive

Figure 3.6 Create Pricing Report: Appearance of the Result List

The system offers the key fields of all previously selected condition tables as selection fields. In the Positioning column, you can choose between three options:

- 1: Page Header of Pricing Report
- 2: Group Header of Pricing Report
- 3: Item Level of Pricing Report

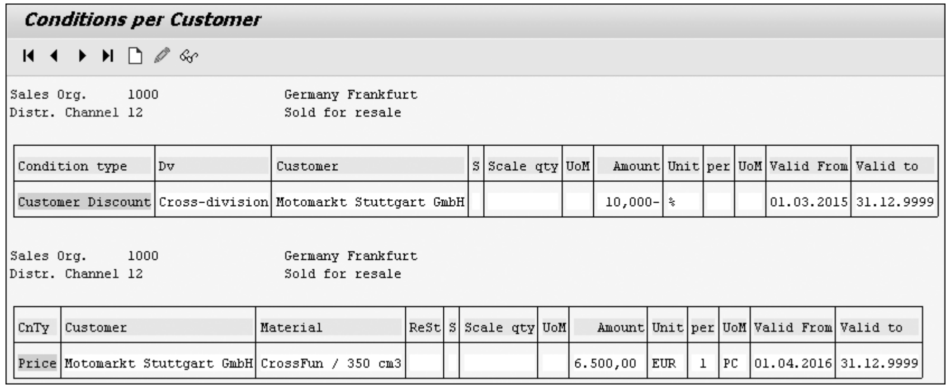
In the Sort column, the order of the fields in a level (page header, group header, or item level) can be set. The sort order does not affect the order of the page header, group header, and item level itself.

In the Text column, you also have three options available:

- 1: Only the key field is displayed
- 2: The key field and the corresponding text are displayed
- 3: Only the corresponding text is displayed

These options apply only to the levels of group header and item level. In contrast, in the page header, the field name and description are always displayed. Due to lack of space, either the field name or the associated text should be displayed at the item level.

In the Selection column, you can deselect some of the selection fields as necessary, though we have not done so in our example. In the Required input column, the Sales Organization field has been marked as a required entry field. Click Save to generate the new pricing report X0. A first test run of the new pricing report shows the result displayed in Figure 3.7.



Conditions per Customer

```
Sales Org.     1000                Germany Frankfurt
Distr. Channel 12                  Sold for resale
```

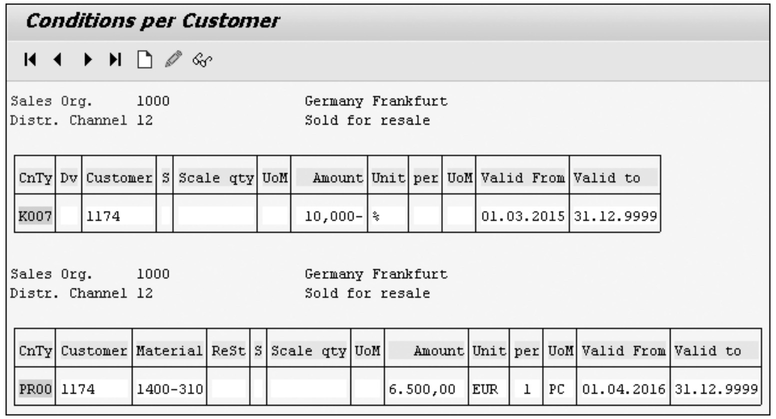
Condition type	Dv	Customer	S	Scale qty	UoM	Amount	Unit	per	UoM	Valid From	Valid to
Customer Discount	Cross-division	Motomarkt Stuttgart GmbH				10,000-	%			01.03.2015	31.12.9999

```
Sales Org.     1000                Germany Frankfurt
Distr. Channel 12                  Sold for resale
```

CnTy	Customer	Material	ReSt	S	Scale qty	UoM	Amount	Unit	per	UoM	Valid From	Valid to
Price	Motomarkt Stuttgart GmbH	CrossFun / 350 cm3					6.500,00	EUR	1	PC	01.04.2016	31.12.9999

Figure 3.7 Conditions per Customer: First Test Run

Since the appearance of the result list is not completely in line with our expectations, we edit the pricing report again in change mode and replace the attributes in the Text column labeled 3 - Only the corresponding text is displayed with the attribute 1 - Only the key field is displayed. A new call of the pricing report X0 shows the result displayed in Figure 3.8.



Conditions per Customer

```
Sales Org.     1000                Germany Frankfurt
Distr. Channel 12                  Sold for resale
```

CnTy	Dv	Customer	S	Scale qty	UoM	Amount	Unit	per	UoM	Valid From	Valid to
K007		1174				10,000-	%			01.03.2015	31.12.9999

```
Sales Org.     1000                Germany Frankfurt
Distr. Channel 12                  Sold for resale
```

CnTy	Customer	Material	ReSt	S	Scale qty	UoM	Amount	Unit	per	UoM	Valid From	Valid to
PR00	1174	1400-310					6.500,00	EUR	1	PC	01.04.2016	31.12.9999

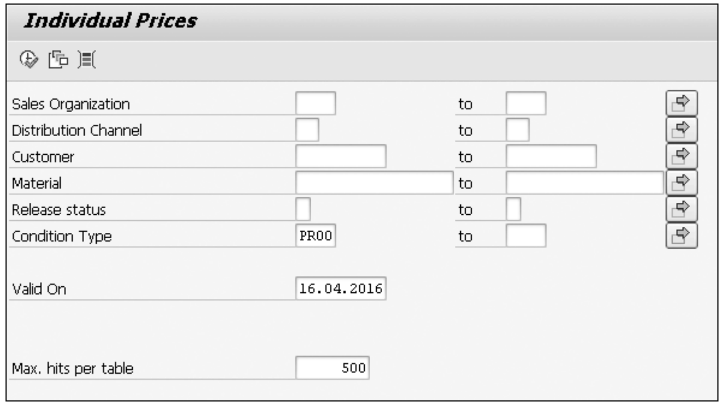
Figure 3.8 Conditions per Customer: Second Test Run

Usually several change rounds will be necessary to customize the options of the result list according to your needs.

3.1.3 Pricing Report for Condition Maintenance via Area Menu

As described in Chapter 2, the pricing reports play an important role in the condition maintenance via area menus. They act as filters, which are assigned to a node of the area menu in the form of a variant of a pricing report (e.g., in area menu COND_AV).

If you want to use a pricing report as a filter in an area menu, no specific settings are required when generating the report. However, the condition maintenance Transactions VK31 to VK34 provide an adapted selection screen that is controlled by them. Therefore, the selection screen shown in Figure 3.1 of pricing report 16 (Individual Prices) looks slightly different when used as a filter in the condition maintenance (see Figure 3.9).



Individual Prices

Sales Organization		to	
Distribution Channel		to	
Customer		to	
Material		to	
Release status		to	
Condition Type	PR00	to	
Valid On	16.04.2016		
Max. hits per table	500		

Figure 3.9 Pricing Report as a Filter in the Condition Maintenance

Pricing reports are by themselves very useful and flexible tools made even more valuable by their use as a filter element in the condition maintenance via area menus.

The latest development in condition lists are the *worklists* introduced for the internal sales representative role, which are presented next.

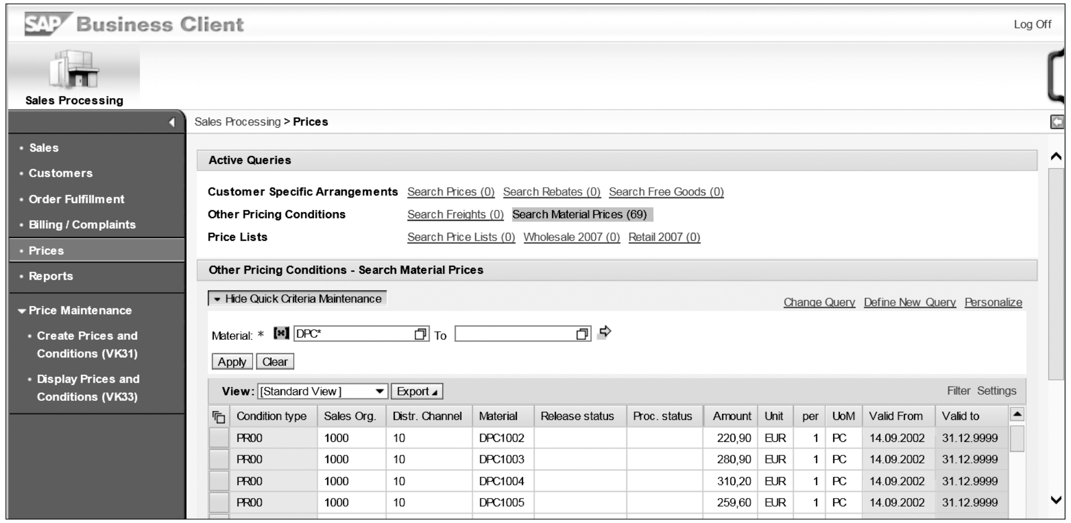
3.2 Worklists for the Internal Sales Representative Role

Since SAP ERP 6.0 EHP 2, the *Internal Sales Representative* business package has been available. To use it, however, you must activate the business function SD_01 (Logistics: S&D Simplification) with Transaction SFW5 (Switch Framework).

In the accompanying supplied sample user role SAP_BPR_INT_SALES_REP_14 (or SAP_SR_INT_SALES_REP_5 later with EHP 5), a personal worklist for condition information is contained. This worklist provides the sales staff with a flexible overview of the different types of conditions as well as extensive search facilities to identify condition master data according to different selection criteria. The individual queries are provided in the form of personal worklists—also called POWLs (personal object worklists). The salesperson receives different lists with the customer-specific agreements such as customer-specific prices, discounts, and free goods. In addition, lists are available with material prices, freight costs, and price lists.

This POWL can be adjusted individually by the user with regard to their use of selection criteria (Change Query), the visibility and the name of the queries (Personalize), and the presentation of the search result (Settings). In addition, users can even create their own queries (Define New Query).

Initially, the functionality of the internal sales representative role was exclusively available via portal usage. Starting with SAP ERP 6.0 EHP 4, there is also the option to use the role with the SAP Business Client. The functionality relating to the worklist for condition information is identical. Figure 3.10 shows the worklist for condition information after accessing the system with the SAP Business Client and choosing menu node Prices.



Sales Processing > **Prices**

Active Queries

Customer Specific Arrangements	Search Prices (0) Search Rebates (0) Search Free Goods (0)
Other Pricing Conditions	Search Freights (0) Search Material Prices (69)
Price Lists	Search Price Lists (0) Wholesale 2007 (0) Retail 2007 (0)

Other Pricing Conditions - Search Material Prices

Material: * DPC* To

View: [Standard View]

Condition type	Sales Org.	Distr. Channel	Material	Release status	Proc. status	Amount	Unit	per	UoM	Valid From	Valid to
PR00	1000	10	DPC1002			220,90	EUR	1	PC	14.09.2002	31.12.9999
PR00	1000	10	DPC1003			280,90	EUR	1	PC	14.09.2002	31.12.9999
PR00	1000	10	DPC1004			310,20	EUR	1	PC	14.09.2002	31.12.9999
PR00	1000	10	DPC1005			259,60	EUR	1	PC	14.09.2002	31.12.9999

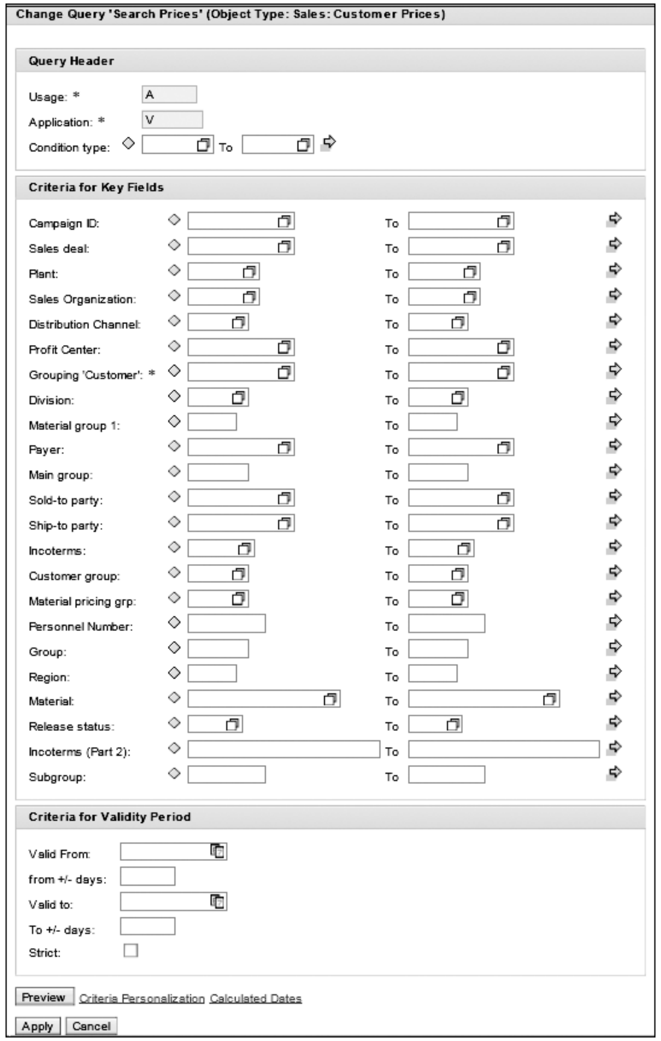
Figure 3.10 Screen of the Internal Sales Representative Role

To ensure that you receive insight into the capabilities of POWL, we will discuss the features of this list in the following sections, using a practical example.

3.2.1 Change Query

The queries in the personal worklist should of course reflect the specific information needs of the user. You can adjust existing queries by using the function Change Query (see Figure 3.10). In our example, we select the query Search Material Prices in the screen area Active Queries and choose the function

Change Query. As a result, a selection screen opens with a large number of possible selection fields, which you can fill in with any combinations (see Figure 3.11).



Change Query 'Search Prices' (Object Type: Sales: Customer Prices)

Query Header

Usage: * A

Application: * V

Condition type: ◇ To

Criteria for Key Fields

Campaign ID:	◇	To
Sales deal:	◇	To
Plant:	◇	To
Sales Organization:	◇	To
Distribution Channel:	◇	To
Profit Center:	◇	To
Grouping 'Customer': *	◇	To
Division:	◇	To
Material group 1:	◇	To
Payer:	◇	To
Main group:	◇	To
Sold-to party:	◇	To
Ship-to party:	◇	To
Incoterms:	◇	To
Customer group:	◇	To
Material pricing grp:	◇	To
Personnel Number:	◇	To
Group:	◇	To
Region:	◇	To
Material:	◇	To
Release status:	◇	To
Incoterms (Part 2):	◇	To
Subgroup:	◇	To

Criteria for Validity Period

Valid From:

from +/- days:

Valid to:

To +/- days:

Strict: □

Figure 3.11 Selection Fields of the Search Prices Query

The available selection fields are defined in the configuration of the underlying POWL. This configuration can be reached via the menu path IMG • Sales and Distribution • "Internal Sales Representative" Role • Personal Worklist (New) • Personal Worklist for Condition Information.

The main worklist types for sales employees' work areas are predefined and assigned to SAP ERP Sales and Distribution (SD):

- Sales: Customer Rebates
- Sales: Customer Free Goods
- Sales: Customer Prices
- Sales: Document Prices
- Sales: Freight
- Sales: Material Prices
- Sales: Price List

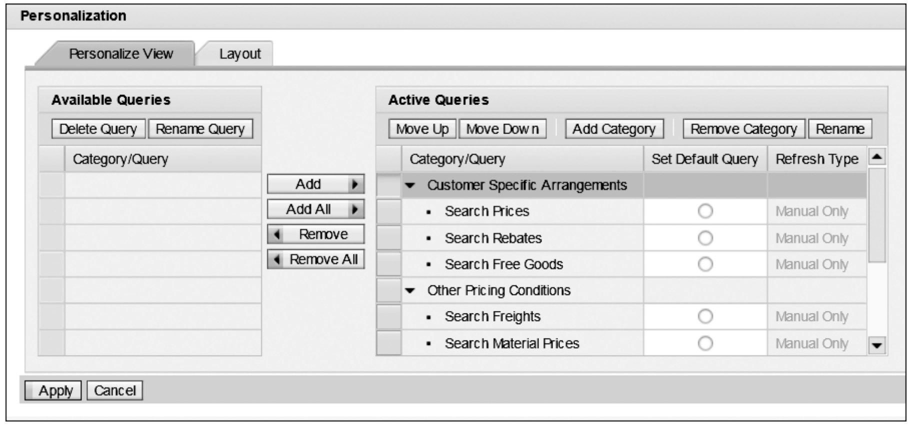
Customer-Specific Worklist Types

In addition to the predefined worklist types, customer-specific worklist types can be created. Nevertheless, for this purpose, Business Add-ins (BAdIs) must be programmed!

As you can see on closer inspection in Figure 3.11, the underlying worklist type for the query Search Prices is the type Sales: Customer Prices.

3.2.2 Personalize Query

To personalize a query, select the Personalize function in the initial screen of the query (see Figure 3.10). This will open the screen shown in Figure 3.12. You can enable or disable the available queries and design your worklist.



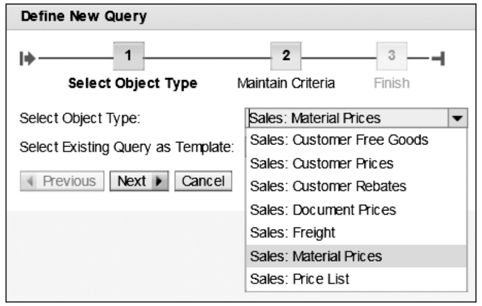
Category/Query	Set Default Query	Refresh Type
Customer Specific Arrangements		
• Search Prices	○	Manual Only
• Search Rebates	○	Manual Only
• Search Free Goods	○	Manual Only
Other Pricing Conditions		
• Search Freights	○	Manual Only
• Search Material Prices	○	Manual Only

Figure 3.12 Personalize Query

3.2.3 Define New Query

To search condition records according to other aspects, you can create new queries. After selecting the Define New Query function, a wizard starts that guides the user in three steps through the creation of the new query.

First, select the object type of the query (see Figure 3.13).



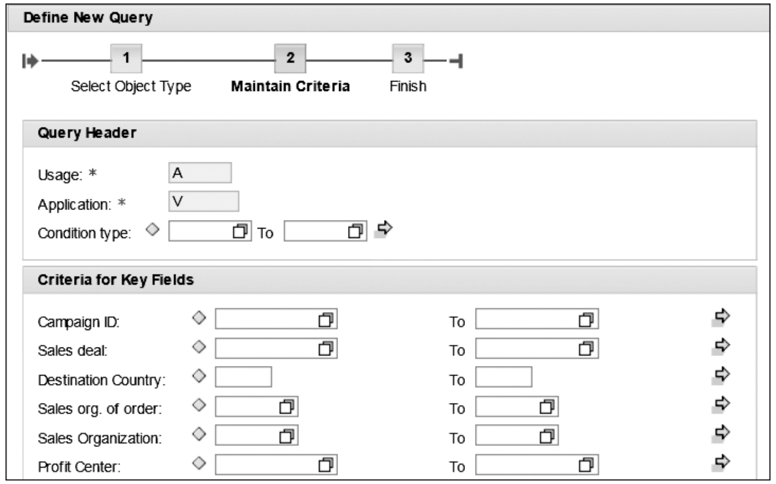
Define New Query

Select Object Type: Sales: Material Prices

Select Existing Query as Template:

Figure 3.13 Define New Query: Step 1

The second step is in the input of the selection criteria (see Figure 3.14). As already mentioned, the selected query type defines these.



Define New Query

Query Header

Usage: * A

Application: * V

Condition type: To

Criteria for Key Fields

Campaign ID: To

Sales deal: To

Destination Country: To

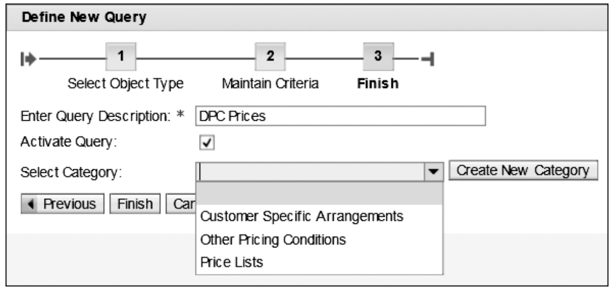
Sales org. of order: To

Sales Organization: To

Profit Center: To

Figure 3.14 Define New Query: Step 2

The third and final step is to specify the description and the category under which the query should appear in the worklist (see Figure 3.15).



Define New Query

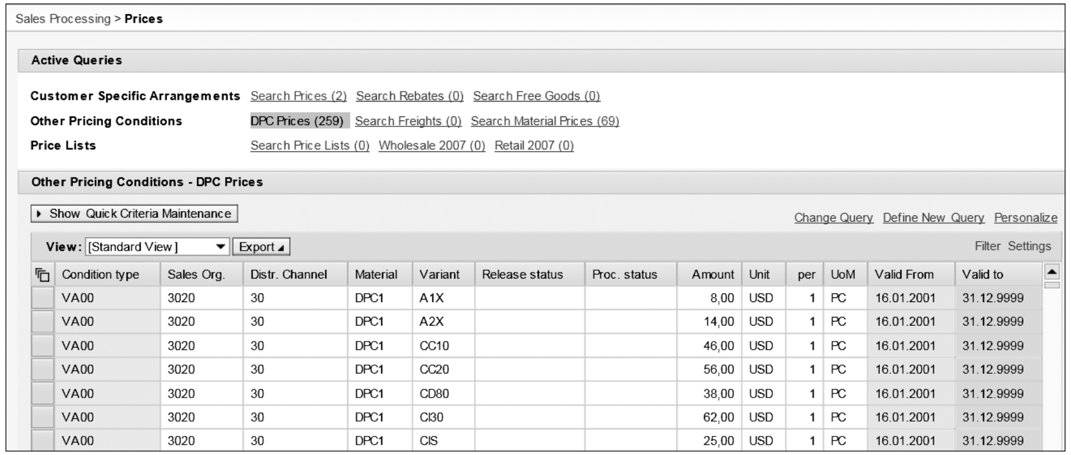
Enter Query Description: * DPC Prices

Activate Query: ☑

Select Category:

Figure 3.15 Define New Query: Step 3

After you click on the Finish button, you can use the new query in the worklist (see Figure 3.16).



Sales Processing > Prices

Active Queries

Customer Specific Arrangements: Search Prices (2) Search Rebates (0) Search Free Goods (0)

Other Pricing Conditions: DPC Prices (259) Search Freights (0) Search Material Prices (69)

Price Lists: Search Price Lists (0) Wholesale 2007 (0) Retail 2007 (0)

Other Pricing Conditions - DPC Prices

View: [Standard View]

Condition type	Sales Org.	Distr. Channel	Material	Variant	Release status	Proc. status	Amount	Unit	per	UoM	Valid From	Valid to
VA00	3020	30	DPC1	A1X			8,00	USD	1	PC	16.01.2001	31.12.9999
VA00	3020	30	DPC1	A2X			14,00	USD	1	PC	16.01.2001	31.12.9999
VA00	3020	30	DPC1	CC10			46,00	USD	1	PC	16.01.2001	31.12.9999
VA00	3020	30	DPC1	CC20			56,00	USD	1	PC	16.01.2001	31.12.9999
VA00	3020	30	DPC1	CD80			38,00	USD	1	PC	16.01.2001	31.12.9999
VA00	3020	30	DPC1	CI30			62,00	USD	1	PC	16.01.2001	31.12.9999
VA00	3020	30	DPC1	CIS			25,00	USD	1	PC	16.01.2001	31.12.9999

Figure 3.16 New Query in the Worklist

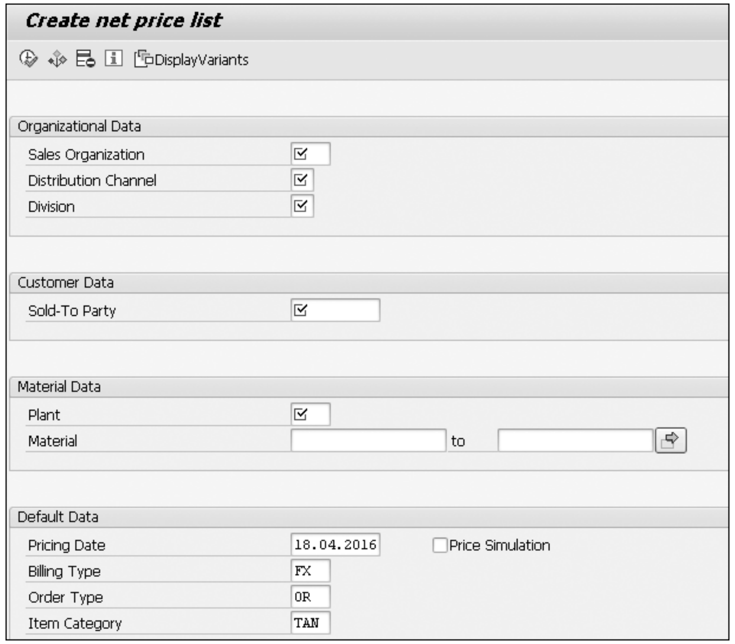
As you can see, it can be worthwhile to use the Internal Sales Representative business package. In addition to the worklist for price data, it contains many other interesting areas, such as the customer cockpit. For more information on this, we recommend a look at the SAP ERP documentation at *http://help.sap.com*.

3.3 Net Price List

The net price list provides price information on a selected set of products for a specific customer. The net prices cannot simply be read from a master data table, but instead must be calculated. For this purpose, an invoice is simulated (using function module GN_INVOICE_CREATE) and a corresponding pricing run is performed. Therefore, in the selection screen of the net price list, all information must be entered that is necessary for a specific pricing procedure determination: sales area, sales document type (from which the Document Pricing Procedure field is derived), and the customer (from which the Customer Pricing Procedure field is derived).

The net price list can be accessed via the menu path SAP Easy Access • Logistics • Sales and Distribution • Master Data • Conditions • List • Net Price List (Transaction V_NL).

A selection screen is displayed (see Figure 3.17), and the previously mentioned selection criteria must be entered for the pricing procedure determination.


Create net price list

Organizational Data

Sales Organization	☑
Distribution Channel	☑
Division	☑

Customer Data

Sold-To Party	☑

Material Data

Plant	☑		
Material		to	

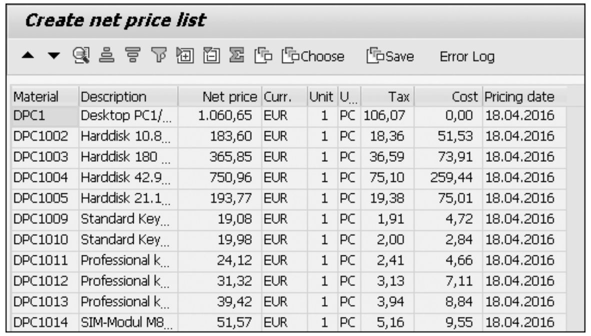
Default Data

Pricing Date	18.04.2016	□ Price Simulation
Billing Type	FX	
Order Type	OR	
Item Category	TAN	

Figure 3.17 Net Price List: Selection Screen (Transaction V_NL)

You also need the Plant field to identify the country of departure (impact on the tax determination). The Price Simulation attribute determines whether conditions with release status B – Released for price simulation or C - Released for planning and price simulation are to be taken into account.

As a result, an SAP List Viewer (ALV) is displayed with a selected set of fields that can be edited with the usual procedure in the ALVs (menu path Settings • Layout) (see Figure 3.18).


Create net price list

Material	Description	Net price	Curr.	Unit	U...	Tax	Cost	Pricing date
DPC1	Desktop PC1/...	1.060,65	EUR	1	PC	106,07	0,00	18.04.2016
DPC1002	Harddisk 10.8...	183,60	EUR	1	PC	18,36	51,53	18.04.2016
DPC1003	Harddisk 180 ...	365,85	EUR	1	PC	36,59	73,91	18.04.2016
DPC1004	Harddisk 42.9...	750,96	EUR	1	PC	75,10	259,44	18.04.2016
DPC1005	Harddisk 21.1...	193,77	EUR	1	PC	19,38	75,01	18.04.2016
DPC1009	Standard Key...	19,08	EUR	1	PC	1,91	4,72	18.04.2016
DPC1010	Standard Key...	19,98	EUR	1	PC	2,00	2,84	18.04.2016
DPC1011	Professional k...	24,12	EUR	1	PC	2,41	4,66	18.04.2016
DPC1012	Professional k...	31,32	EUR	1	PC	3,13	7,11	18.04.2016
DPC1013	Professional k...	39,42	EUR	1	PC	3,94	8,84	18.04.2016
DPC1014	SIM-Modul M8...	51,57	EUR	1	PC	5,16	9,55	18.04.2016

Figure 3.18 Net Price List: Result Screen

The net price list, which can also contain information such as transfer prices, is primarily an internal information tool for the department. If such a net price list is forwarded to the customer, of course, that information must be removed from the list. Since it is an ALV, a personalization of the columns displayed is easily possible by choosing menu path Settings • Layout • Current.

3.4 Performance-Optimized Price List

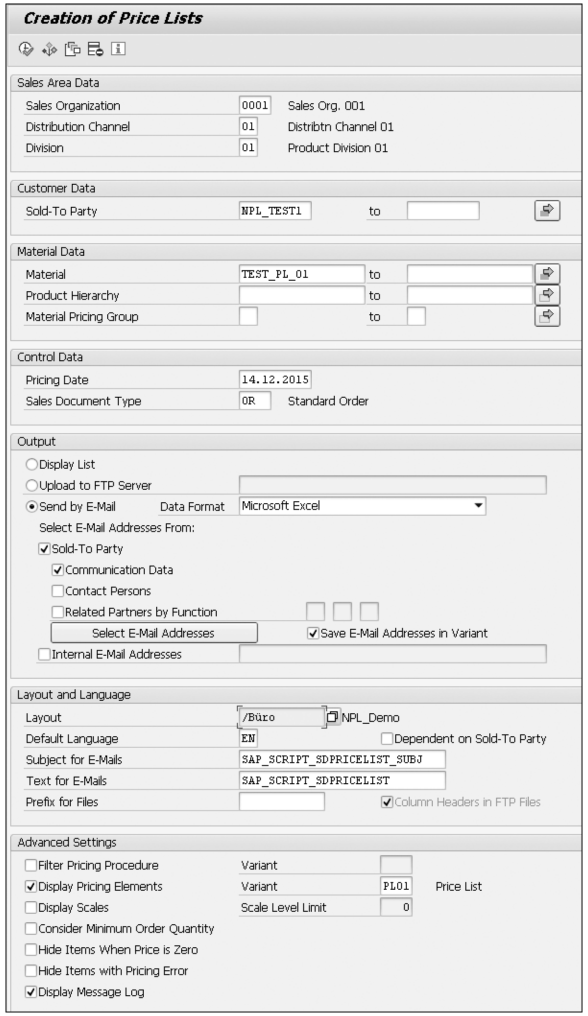
Starting with SAP ERP 6.0 EHP 7, Support Package 3, a performance-optimized price list is offered with business function LOG_SD_PRICE_LIST and enhanced with LOG_SD_PRICE_LIST_2 (Support Package 4). Unlike the original net price list, the performance-optimized price list can handle multiple customers at a time and has, in general, more options. Its purpose is not so much to provide information to the department (although this is of course still possible) but to create lists that, for example, can be sent automatically and on a regular basis to your customers. The technical realization of this new list is quite different and no longer uses any document simulation. We will comment on this later when discussing the various possibilities of customer enhancements in Section 3.4.2, Section 3.4.5, and Section 3.4.6.

Let us begin with the creation of an optimized price list and then have a closer look on the configuration and enhancement possibilities.

3.4.1 Creation of Price Lists

If the relevant business functions are switched on, the optimized price list can be accessed via the menu path SAP Easy Access • Logistics • Sales and Distribution • Master Data • Conditions • List • Net Price List (Transaction V_NLN), thereby replacing the original list that you can still invoke with Transaction V_NL.

At first glance, the upper part of the selection screen in Figure 3.19 looks very similar to the traditional price list from Figure 3.17.


Creation of Price Lists

Sales Area Data
Sales Organization: 0001 Sales Org. 001
Distribution Channel: 01 Distribtn Channel 01
Division: 01 Product Division 01

Customer Data
Sold-To Party: NPL_TEST1 to

Material Data
Material: TEST_PL_01 to
Product Hierarchy: to
Material Pricing Group: to

Control Data
Pricing Date: 14.12.2015
Sales Document Type: OR Standard Order

Output
○ Display List
○ Upload to FTP Server
● Send by E-Mail Data Format: Microsoft Excel
Select E-Mail Addresses From:
☑ Sold-To Party
☑ Communication Data
□ Contact Persons
□ Related Partners by Function
Select E-Mail Addresses ☑ Save E-Mail Addresses in Variant
□ Internal E-Mail Addresses

Layout and Language
Layout: /Büro NPL_Demo
Default Language: EN □ Dependent on Sold-To Party
Subject for E-Mails: SAP_SCRIPT_SDPRICELIST_SUBJ
Text for E-Mails: SAP_SCRIPT_SDPRICELIST
Prefix for Files: ☑ Column Headers in FTP Files

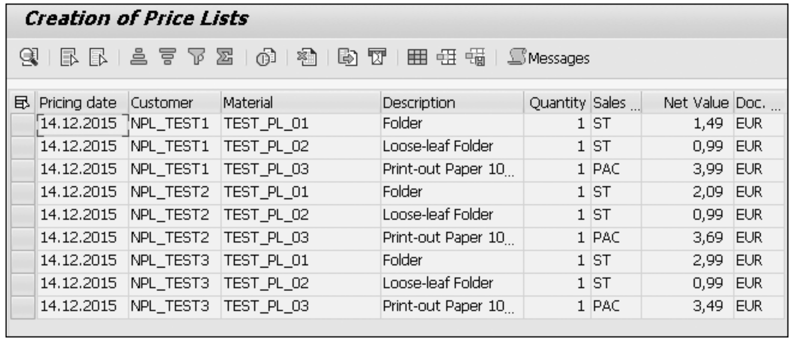
Advanced Settings
□ Filter Pricing Procedure Variant:
☑ Display Pricing Elements Variant: PL01 Price List
□ Display Scales Scale Level Limit: 0
□ Consider Minimum Order Quantity
□ Hide Items When Price is Zero
□ Hide Items with Pricing Error
☑ Display Message Log

Figure 3.19 Selection Screen of the Performance Optimized Price List

If you look closer, you will see that you can now select several customers at once and that the selection of the products is more flexible. Net prices are calculated for each possible customer/material combination.

The lower part of the selection screen offers several new options. There are three radio buttons in the Output section of the selection screen, as follows:

- Display List
  If you choose this button, the prices will be displayed on your screen in the form of one ALV, including all calculated customer/material combinations. You can personalize the list to your needs, create different layouts, filter, etc. You can enter an existing layout in the Layout and Language section. We will describe later how to control and enhance the available fields in the output list. A possible result could look like the one shown in Figure 3.20.


Creation of Price Lists

Pricing date	Customer	Material	Description	Quantity	Sales ...	Net Value	Doc. ...
14.12.2015	NPL_TEST1	TEST_PL_01	Folder	1	ST	1,49	EUR
14.12.2015	NPL_TEST1	TEST_PL_02	Loose-leaf Folder	1	ST	0,99	EUR
14.12.2015	NPL_TEST1	TEST_PL_03	Print-out Paper 10...	1	PAC	3,99	EUR
14.12.2015	NPL_TEST2	TEST_PL_01	Folder	1	ST	2,09	EUR
14.12.2015	NPL_TEST2	TEST_PL_02	Loose-leaf Folder	1	ST	0,99	EUR
14.12.2015	NPL_TEST2	TEST_PL_03	Print-out Paper 10...	1	PAC	3,69	EUR
14.12.2015	NPL_TEST3	TEST_PL_01	Folder	1	ST	2,99	EUR
14.12.2015	NPL_TEST3	TEST_PL_02	Loose-leaf Folder	1	ST	0,99	EUR
14.12.2015	NPL_TEST3	TEST_PL_03	Print-out Paper 10...	1	PAC	3,49	EUR

Figure 3.20 Example Output of Performance-Optimized Price List

- Upload to FTP Server
  Alternatively, you can upload the price list as a CSV file to an FTP server. The server address, user, and password are defined in the Customizing of the price list function (IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Selection and Result • Define FTP Settings). Price lists for different customers are stored in different files, with the customer number being part of the filename. You may define a prefix for the files in the Layout and Language section of the report. The system will then put the prefix in front of the filenames that are composed as follows: `<Prefix>_<Customer Number>_<Timestamps>`.

  The files can be stored with or without the column headers (checkbox in the Layout and Language section).

- Send by E-Mail

  It is also possible to send the price lists via e-mail. The report allows you to select the e-mail addresses from different sources such as, for example, the communication data of the customer or the contact person in the customer master data. You can also set the e-mail addresses manually for each customer (the Select E-Mail Addresses button) and store them in a report's variant, or send all lists to one or several internal e-mail addresses. SAP provides the `PROCESS_CUSTOMER_EMAIL` method in the BAdI `BADI_PIQ_SDPRICELIST`, where you can program you own logic if necessary.

  Price lists can be sent in CSV or Microsoft Excel format. For all output options, the system will take the denoted layout from the Layout and Language section into account and send one list per customer. You can specify a default language for all texts or decide to retrieve the language for each customer separately from the sold-to master data record. You can also create language-dependent SAPscript texts for the e-mail header and body in Transaction SO10 and specify them on the selection screen. The texts can also be filled in using the method `PROVIDE_EMAIL_TEXT_NAME` of the BAdI `BADI_PIQ_SDPRICELIST`.

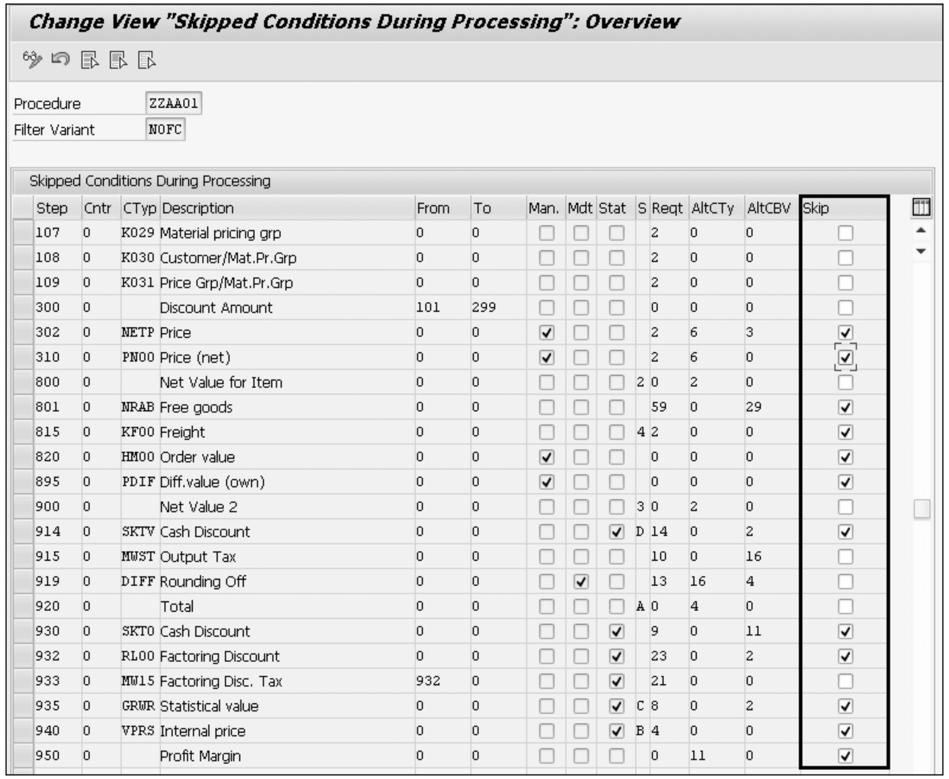
The advanced settings section offers options to influence the result of the price list, as follows:

- Filter Pricing Procedure

  When creating a price list, you may not need all lines of your pricing procedure. Maybe you do not care about the freight cost or you are not interested in the cost conditions. The new price list Customizing offers the definition of *filter variants* for pricing procedures where you mark non-relevant steps. These steps are then ignored or skipped during the price calculation, thus improving performance. You can maintain one or more filters for a pricing procedure here: IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Selection and Result • Define Filter for Pricing Procedure. Just flag the rows in the procedure that you do not need in the price list calculation (see Figure 3.21).

- Display Pricing Elements

  The previous net price list provides net price and tax information but no information about individual price elements or subtotal lines, for example the overall discount or a special sales deal. In a similar way to filtering some lines from the pricing procedure, you may define *display variants* where you mark those lines in the procedure that will be available for output in the price list result.



Change View "Skipped Conditions During Processing": Overview

Procedure: ZZAA01
Filter Variant: NOFC

Skipped Conditions During Processing

Step	Cntr	CTyp	Description	From	To	Man.	Mdt	Stat	S	Reqt	AltCTy	AltCBV	Skip
107	0	K029	Material pricing grp	0	0	□	□	□		2	0	0	□
108	0	K030	Customer/Mat.Pr.Grp	0	0	□	□	□		2	0	0	□
109	0	K031	Price Grp/Mat.Pr.Grp	0	0	□	□	□		2	0	0	□
300	0		Discount Amount	101	299	□	□	□		0	0	0	□
302	0	NETP	Price	0	0	☑	□	□		2	6	3	☑
310	0	PN00	Price (net)	0	0	☑	□	□		2	6	0	☑
800	0		Net Value for Item	0	0	□	□	□	2	0	2	0	□
801	0	NRAB	Free goods	0	0	□	□	□		59	0	29	☑
815	0	KF00	Freight	0	0	□	□	□	4	2	0	0	☑
820	0	HM00	Order value	0	0	☑	□	□		0	0	0	☑
895	0	PDIF	Diff.value (own)	0	0	☑	□	□		0	0	0	☑
900	0		Net Value 2	0	0	□	□	□	3	0	2	0	□
914	0	SKTV	Cash Discount	0	0	□	□	☑	D	14	0	2	☑
915	0	MWST	Output Tax	0	0	□	□	□		10	0	16	□
919	0	DIFF	Rounding Off	0	0	□	☑	□		13	16	4	□
920	0		Total	0	0	□	□	□	A	0	4	0	□
930	0	SKTO	Cash Discount	0	0	□	□	☑		9	0	11	☑
932	0	RL00	Factoring Discount	0	0	□	□	☑		23	0	2	☑
933	0	MW15	Factoring Disc. Tax	932	0	□	□	☑		21	0	0	□
935	0	GRWR	Statistical value	0	0	□	□	☑	C	8	0	2	☑
940	0	VPRS	Internal price	0	0	□	□	☑	B	4	0	0	☑
950	0		Profit Margin	0	0	□	□	□		0	11	0	☑

Figure 3.21 Definition of a Filter Variant: Rows in the Pricing Procedure Irrelevant for the Net Price List Result Are Marked to be Skipped

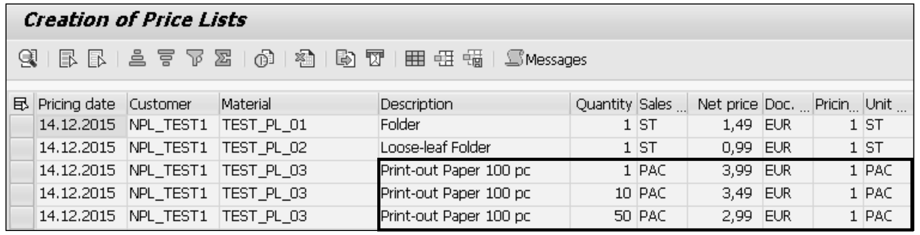
  The relevant Customizing is located in IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Selection and Result • Define Pricing Elements Display. We will explain the exact setting needed to make the flagged elements part of the output list in Section 3.4.6.

- Display Scales

  The performance-optimized price list supports the display of scale information in the result. If you flag the Display Scales checkbox, the system will evaluate all scales involved in the calculation of the net price and will deduce the quantities where it expects the effective net price to change. For these quantities, the system recalculates the price and adds corresponding lines to the result (see Figure 3.22). You can thereby inform your customer about the opportunity to get better prices if they purchase higher quantities.

The number of additional quantities can be limited by the Scale Level Limit field. If you leave it empty, the price list will calculate and display all relevant quantities. If you enter "4", for example, you will receive up to four additional lines or five in total.

The net price list supports any mixture of value, quantity, weight, and volume scales and has a from-scale notation in the output; i.e., the listed price is valid from the indicated sales quantity upwards. Graduated scales are not permitted since they result in different prices for each sales quantity. Scale-base formulas may interfere with the calculation of the sales quantities in an unpredictable way and are similarly not allowed. However, BAdI `BADI_PRC_NPL_SCALES` allows you to overrule these restrictions and to give your own set of sales quantities to be calculated.



Pricing date	Customer	Material	Description	Quantity	Sales ...	Net price	Doc. ...	Pricin...	Unit ...
14.12.2015	NPL_TEST1	TEST_PL_01	Folder	1	ST	1,49	EUR	1	ST
14.12.2015	NPL_TEST1	TEST_PL_02	Loose-leaf Folder	1	ST	0,99	EUR	1	ST
14.12.2015	NPL_TEST1	TEST_PL_03	Print-out Paper 100 pc	1	PAC	3,99	EUR	1	PAC
14.12.2015	NPL_TEST1	TEST_PL_03	Print-out Paper 100 pc	10	PAC	3,49	EUR	1	PAC
14.12.2015	NPL_TEST1	TEST_PL_03	Print-out Paper 100 pc	50	PAC	2,99	EUR	1	PAC

Figure 3.22 Output of the Price List with Scales Displayed

- Consider Minimum Order Quantity
  Usually the price list will calculate the net price and value for the quantity 1 in sales units of measure. If you mark this checkbox, the prices will be calculated for the minimum order quantity instead.
- Hide Item When Price is Zero
  Customer/material combinations with a zero net price will be excluded from the result list.
- Hide Items with Pricing Error
  Customer/material combinations with errors in the price calculation will be excluded from the result list.
- Display Message Log
  We will explain in the next section how the necessary data for the pricing call is gathered. Especially in the setup and test phases, the message log can give you valuable hints if the required Customizing settings for the data retrieval are not complete.

3.4.2 Customizing and Customer Enhancements

We already mentioned that the performance-optimized price list does not use any kind of business document simulation, so user exits, like those found in the sales order or billing document, are not called. Instead, the price list coding analyzes the access sequences, user exits, formulas, and requirements used in the relevant pricing procedures to determine the actual list of required fields for the pricing communication structures KOMK and KOMP. Only the relevant fields are then determined, with as few database accesses as possible.

While the system can identify fields needed in access sequences reliably, this is not possible for fields used in formulas or requirements. That is why this information is stored in a dedicated customizing table. To allow an effective and flexible bundling of database selects, the necessary table accesses for data retrieval are also stored in customizing tables in form of metadata.

If you use customer-specific fields in access sequences, user exits, formulas, or requirements, you need to include the same information (which fields are needed and how they are determined) in the customer areas in Customizing.

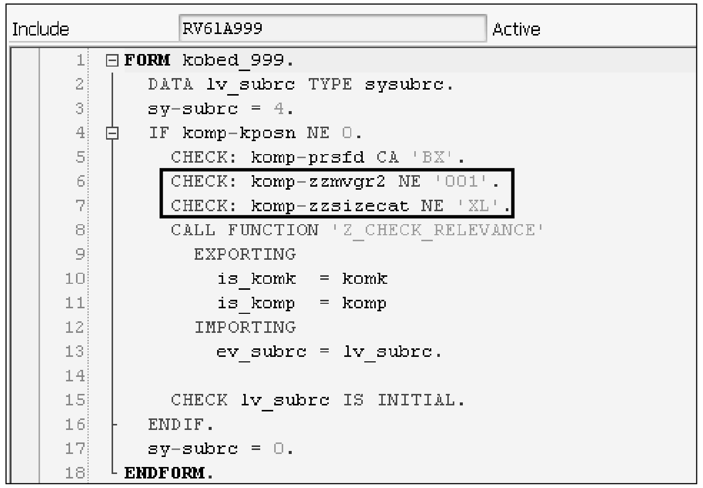
Note

We will now explain the necessary settings with an example. In doing so, we have to assume some technical knowledge and need to delve into some details and enhancement concepts described later in the book, especially topics covered in Chapters 9 and 11. If you are more of a novice in pricing, you may want to skip the following sections for now and continue with Chapter 4.

You will find all necessary activities under the node IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists.

3.4.3 Registration of Fields

Let's assume that we have added the fields ZZMVGR1, ZZMVGR2, ZZSIZECAT, and ZZPROMO to the item communication structure of pricing KOMP. ZZMVGR1 and ZZMVGR2 correspond to the fields MVGR1 (material group 1) and MVGR2 (material group 2) from the sales data in the material master record. ZZSIZECAT is stored in our own table `ZZTSIZECAT` with the material number as the key field. ZZPROMO is derived by some dedicated business logic. We assume that we use ZZMVGR1 in one of our condition accesses, while ZZMVGR2, ZZSIZECAT, and ZZPROMO are used only in our requirement 999 (see Figure 3.23). ZZPROMO is used inside function module `Z_CHECK_RELEVANCE`.



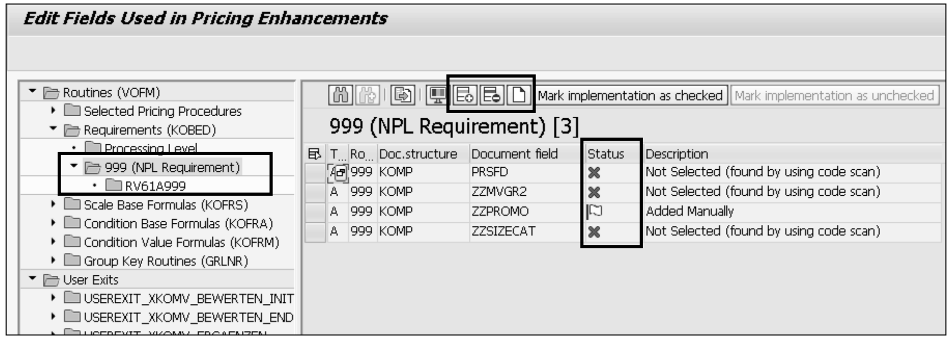
```
FORM kobed_999.
  DATA lv_subrc TYPE sysubrc.
  sy-subrc = 4.
  IF komp-kposn NE 0.
    CHECK: komp-prsfd CA 'BX'.
    CHECK: komp-zzmvgr2 NE '001'.
    CHECK: komp-zzsizecat NE 'XL'.
    CALL FUNCTION 'Z_CHECK_RELEVANCE'
      EXPORTING
        is_komk  = komk
        is_komp  = komp
      IMPORTING
        ev_subrc = lv_subrc.

    CHECK lv_subrc IS INITIAL.
  ENDIF.
  sy-subrc = 0.
ENDFORM.
```

Figure 3.23 Requirement 999 Using Fields ZZMVGR2, ZZSIZECAT, and ZZPROMO

The performance-optimized price list tries to identify all needed fields for a specific price calculation. Therefore, it scans all access sequences of the pricing procedure involved and thus becomes aware of ZZMGR1. However, fields required in the execution of user exits, customer-specific formulas, or requirements must be explicitly published in Customizing. To do so, choose the IMG activity IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Edit Fields Used in Pricing Enhancements.

On the entry screen, we restrict the view to the relevant pricing procedure and decide that we want to see only customer-developed enhancements. We navigate to requirement 999. A double-click on include `RV61A999` would lead us to the coding, but we double-click the node 999 instead.

We now have the list of fields automatically found by means of a simple code scans: ZZMVGR2 and ZZSIZECAT. However, the field KOMP-ZZPROMO is "hidden" behind a function module call and hence was missed. Therefore, we register it manually by clicking the Add Field Manual button and entering the necessary data. You can see the result in Figure 3.24, where the found fields have not yet been confirmed and the ZZPROMO field has been added manually.

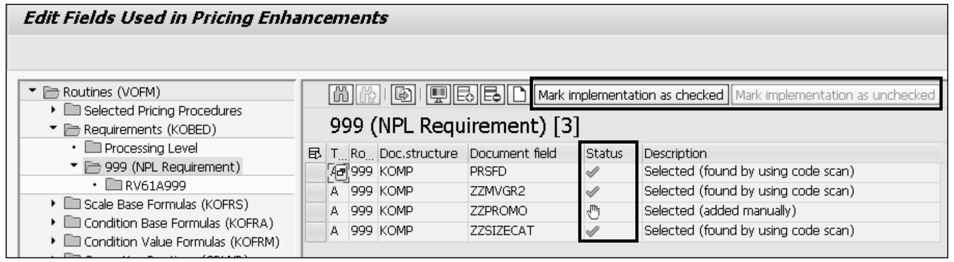

Edit Fields Used in Pricing Enhancements
Routines (VOFM)
Selected Pricing Procedures
Requirements (KOBED)
Processing Level
999 (NPL Requirement)
RV61A999
Scale Base Formulas (KOFRS)
Condition Base Formulas (KOFRA)
Condition Value Formulas (KOFRM)
Group Key Routines (GRLNR)
User Exits
USEREXIT_XKOMV_BEWERTEN_INIT
USEREXIT_XKOMV_BEWERTEN_END
Mark implementation as checked
Mark implementation as unchecked

999 (NPL Requirement) [3]

T...	Ro...	Doc.structure	Document field	Status	Description
A	999	KOMP	PRSFD	✗	Not Selected (found by using code scan)
A	999	KOMP	ZZMVGR2	✗	Not Selected (found by using code scan)
A	999	KOMP	ZZPROMO	⚐	Added Manually
A	999	KOMP	ZZSIZECAT	✗	Not Selected (found by using code scan)

Figure 3.24 Result of the Generic Code Scan for Requirement 999 Searching for Relevant Fields

In the second step, we have to select all fields that must be filled in a price list run (there may be fields that are not relevant) and confirm them explicitly. For this, we select the field(s) and click the Select Fields button ⊕. The status of the fields then changes to Selected (Figure 3.25). You can remove fields that are not relevant for the price lists calculation again with the Deselect button ⊖.


Edit Fields Used in Pricing Enhancements
Routines (VOFM)
Selected Pricing Procedures
Requirements (KOBED)
Processing Level
999 (NPL Requirement)
RV61A999
Scale Base Formulas (KOFRS)
Condition Base Formulas (KOFRA)
Condition Value Formulas (KOFRM)
Mark implementation as checked
Mark implementation as unchecked

999 (NPL Requirement) [3]

T...	Ro...	Doc.structure	Document field	Status	Description
A	999	KOMP	PRSFD	✓	Selected (found by using code scan)
A	999	KOMP	ZZMVGR2	✓	Selected (found by using code scan)
A	999	KOMP	ZZPROMO	✋	Selected (added manually)
A	999	KOMP	ZZSIZECAT	✓	Selected (found by using code scan)

Figure 3.25 Fields Have Been Selected as Relevant

Since the automatic code scan is quite basic, you should always check the system's proposal for your routines and user exits.

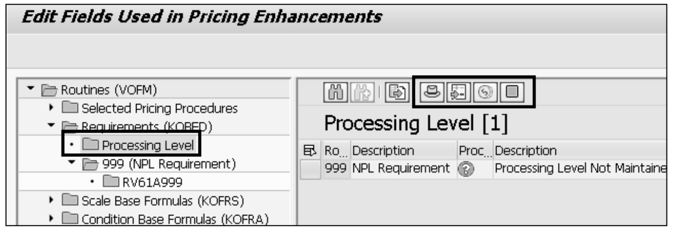
After reviewing and completing all entries, you finalize and document your settings by clicking on the Mark implementation as checked button. Only now will the net price list take your settings into account.

Note

You can undo your settings any time by clicking the Mark implementation as unchecked button. Manually added fields will automatically be removed.

In addition to specifying which fields are required in your pricing exits, you may also specify a *processing level* for your requirements (Figure 3.26). If, for example, a requirement uses only fields from the header communication structure KOMK (so-called header fields), it is usually sufficient to test it only once per pricing header and buffer the result for the following items.

The available processing levels are Header, Item, Condition Type (the hat, list, and coin icons, respectively) and Standard Processing Level (the green square icon). The Standard Processing Level corresponds to an unmaintained processing level. Here nothing is buffered and the requirement is processed each time it's necessary. In simple cases, this is often faster than writing and reading a buffer. That is why SAP has not maintained any level for many standard requirements.


Edit Fields Used in Pricing Enhancements
Routines (VOFM)
Selected Pricing Procedures
Requirements (KOBED)
Processing Level
999 (NPL Requirement)
RV61A999
Scale Base Formulas (KOFRS)
Condition Base Formulas (KOFRA)
Processing Level [1]

Ro...	Description	Proc...	Description
999	NPL Requirement	?	Processing Level Not Maintaine

Figure 3.26 Setting the Processing Level of a Requirement

When setting the processing level, you need to be aware that, for performance reasons, the flow logic of pricing is slightly different in the price list processing. The condition accesses are not performed on each item separately in the pricing function module, but for all items together prior to pricing. Consequently, the call of the requirements is brought forward too, and takes place outside of pricing. Hence, if you do more in your requirement than simply check the relevance of a condition, you should assign the standard level or no level at all. By doing so, you ensure that the requirement is called again at the usual place. An example of such a routine is the standard requirement 62, which fills the internal table XKOMPLOOP.

3.4.4 Define the Data Determination

We have already mentioned that the new price list tries to bundle database accesses. This applies in particular when filling the communication structures in pricing. Thus, in order to enable the system to identify collectable accesses in a generic way, the necessary information is stored in the form of metadata. It is important that you understand how the price list determines its data since you must provide the same information in your own fields. For this purpose, we will have a closer look now at the definition of the table accesses and the corresponding source fields (including customer-specific fields).

Define Table Access

Setting up the metadata consists of two steps. First, you need to specify how a database table will be accessed. This happens in IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Price Inquiry • Define Table Access.

In Figure 3.27 you will see that for the table containing the sales data of the customer master KNVV, three DB or *table aliases* are defined. For table KNVV, three different aliases are specified: KNVV, KNVV_RG, and KNVV_WE. Each of them accesses the table with different fields.

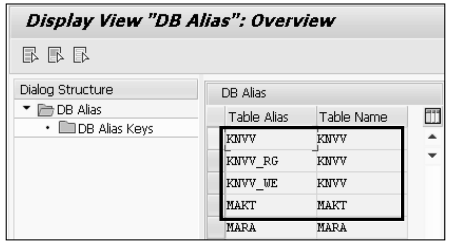
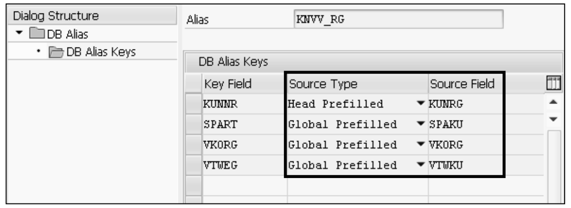


Table Alias	Table Name
KNVV	KNVV
KNVV_RG	KNVV
KNVV_WE	KNVV
MAKT	MAKT
MARA	MARA

Figure 3.27 Definition of Table Alias

If we navigate to the DB Alias Key node, we realize that the three aliases differ in their access fields. The alias KNVV accesses the table with the contents of the source fields KUNNR, SPAKU, VKORG, and VTWKU, reading data from the sold-to party. The alias KNCC_RG, on the other hand, reads data from the payer, accessing the same table as the KUNRG, SPAKU, VKORG, and VTWKU fields (Figure 3.28). The DB alias thus defines the table and the Where clause of the required Select statement.



Alias KNVV_RG

Key Field	Source Type	Source Field
KUNNR	Head Prefilled	KUNRG
SPART	Global Prefilled	SPAKU
VKORG	Global Prefilled	VKORG
VTWEG	Global Prefilled	VTWKU

Figure 3.28 The DB Alias KNVV_RG Accesses the Sales Data of the Payer

What does the Source Type in Figure 3.28 represent? It defines where the source field comes from, or technically speaking, in what internal structure the field resides. We distinguish the following fields:

- Global Prefilled data is relevant for the whole price list, such as organizational data or the price date
- Head Prefilled data is usually derived from the customer and relevant for all products
- Item Prefilled data is usually derived from the material
- HeadItem Prefilled data depends on customer and material combination
- Direct Value is a constant that can be set in Customizing

Simply put, prefilled data is data whose determination is hard-coded in the price list. It is also possible to preempt your own data via a BAdI implementation. In this case, you have to choose the additional data source types (Global, Head, Item, or HeadItem).

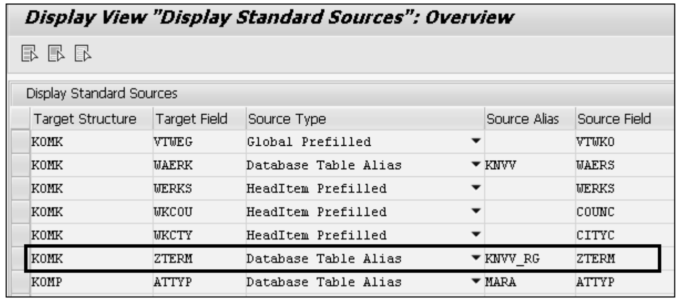
Finally, there is *caller data*. Roughly speaking, this is additional data coming from the outside, especially if the price determination is called outside the net price list via the dedicated function `PIQ_CALCULATE`. We will come back to this very technical detail in Section 3.4.8. For the moment, it is sufficient to know that the F4 help on the source field will provide you with all possible fields for the selected source type.

To summarize, the DB alias `KNVV_RG` can be translated into as follows:

```
select <?> from KNVV into <?> where KUNNR = (Head Prefilled)-KUNRG.
```

Maintain Sources for Communication Structures

In IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Price Inquiry • Sources for Communication Structures we define how the fields relevant for pricing are determined in detail. In Figure 3.29 you can see the fields of the header communication structure KOMK. The distribution channel VTWEG is transferred from prefilled data. Other fields like the payment terms ZTERM are determined using a DB alias. Let's have a closer look at the ZTERM field. The line you see in Figure 3.29, together with the definition of the DB alias KNVV_RG, means that the field ZTERM is filled from the field ZTERM from table `KNVV` using the payer KUNRG to access the table, i.e., the payment terms are read from the sales area data of the payer.



Display View "Display Standard Sources": Overview

Display Standard Sources

Target Structure	Target Field	Source Type	Source Alias	Source Field
KOMK	VTWEG	Global Prefilled		VTWKO
KOMK	WAERK	Database Table Alias	KNVV	WAERS
KOMK	WERKS	HeadItem Prefilled		WERKS
KOMK	WKCOU	HeadItem Prefilled		COUNC
KOMK	WKCTY	HeadItem Prefilled		CITYC
KOMK	ZTERM	Database Table Alias	KNVV_RG	ZTERM
KOMP	ATTYP	Database Table Alias	MARA	ATTYP

Figure 3.29 Sources of the Fields from the Pricing Communication Structures

Together with the DB alias definition, the following `select` will be performed:

```
select ZTERM from KNVV into KOMK-ZTERM where KUNNR = (Head Prefilled)-KUNRG.
```

After evaluating all metadata, the price list can optimize the access and read all required fields from a database table together.

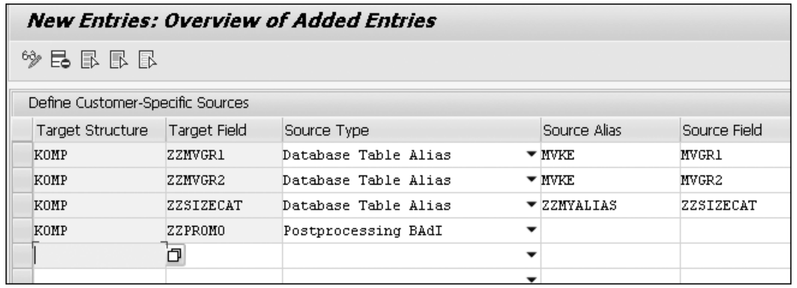
Available Fields with Standard Sources

The price list report does not yet support all fields from the communication structure. You can find out which fields are available by displaying the standard sources in the IMG. If the required field is not listed there, you need to add a corresponding entry in the customer section. Customer entries will always overrule standard settings.

Add Customer Fields

It is time to come back to our example. With what you have just learned, you can now enter the necessary data to determine the fields.

ZZMVGR1 and ZZMVGR2 correspond to fields MVGR1 and MVGR2 from table `MVKE`. We quickly find a suitable DB alias: Alias MVKE reads from table `MVKE` with material number and organizational data. Therefore, all you need to tell the system is that you want to select material groups 1 and 2 and have them transferred into the fields ZZMVGR1 and ZZMVGR2 (see Figure 3.30).



New Entries: Overview of Added Entries

Define Customer-Specific Sources

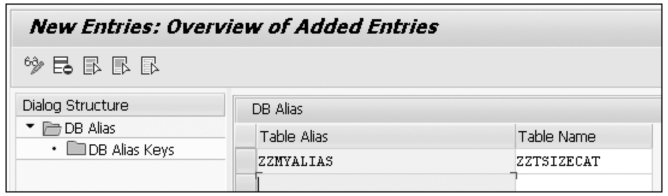
Target Structure	Target Field	Source Type	Source Alias	Source Field
KOMP	ZZMVGR1	Database Table Alias	MVKE	MVGR1
KOMP	ZZMVGR2	Database Table Alias	MVKE	MVGR2
KOMP	ZZSIZECAT	Database Table Alias	ZZMYALIAS	ZZSIZECAT
KOMP	ZZPROMO	Postprocessing BAdI		

Figure 3.30 Define the Sources of Customer Fields

Table Alias

All data derived by the same table alias is selected together. You should therefore always check if you can reuse an existing DB alias to avoid additional, unnecessary database accesses.

We store the field ZZSIZECAT field in our own table so we need to create a new DB alias first. We define that we want to access table ZZTSIZECAT (Figure 3.31) with the prefilled material number (Figure 3.32).



New Entries: Overview of Added Entries

DB Alias

Table Alias	Table Name
ZZMYALIAS	ZZTSIZECAT

Figure 3.31 Creating an DB Alias for Accessing Table ZZTSIZECAT

Next, we need to request the selection of field ZZSIZECAT. Go back to the activity IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Price Inquiry • Define Sources for Pricing Communication Structure and enter the third line shown in Figure 3.30.

We have assumed that our last field ZZPROMO cannot be determined using a simple select. For such cases, SAP provides the BAdI BADI_PIQ_PREPARE, where you can program your own logic. We nevertheless maintain an entry in Customizing for ZZPROMO with the source type Postprocessing BAdI (see Figure 3.30). This will prevent the system from issuing warnings indicating that the source of the field is unknown.

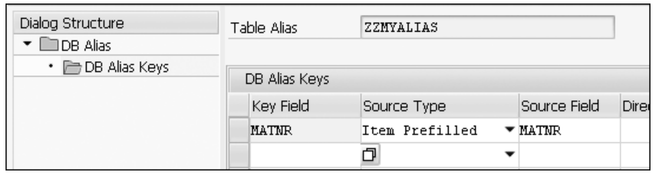


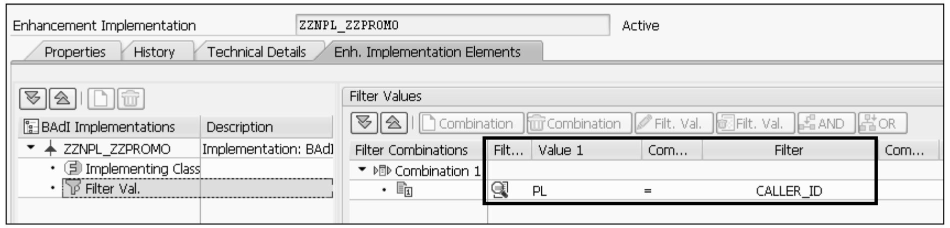
Table Alias ZZMYALIAS

DB Alias Keys

Key Field	Source Type	Source Field	Dire
MATNR	Item Prefilled	MATNR	

Figure 3.32 Defining How to Access Table ZZTSIZECAT

You find BAdI BADI_PIQ_PREPARE in IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Business Add-Ins (BAdIs) • BAdI: Price Inquiry Preparation. As mentioned, the underlying *price inquiry* function can run in different scenarios. Each scenario has a specific caller ID (see Figure 3.33) that defines how to access table ZZTSIZECAT.



Enhancement Implementation ZZNPL_ZZPROMO Active

BAdI Implementations	Description
ZZNPL_ZZPROMO	Implementation: BAdI
Implementing Class	
Filter Val.	

Filter Values

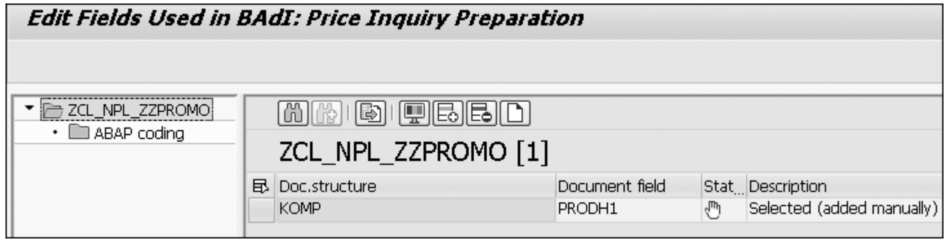
Filter Combinations	Filt...	Value 1	Com...	Filter	Com...
Combination 1					
		PL	=	CALLER_ID	

Figure 3.33 BAdI Implementations Need Filter Value PL to be Relevant for Net Price List Calculations

The method ADAPT_KOMK_KOMP_DATA substantially corresponds to the known user exits in the various documents and allows you to change the content of the pricing communication structures. You will find some tips how to reuse your user exit coding in SAP Note 2019242. Looking at the signature of the method, you can see that apart from the header and item table, there is an additional table called CT_KOMP_HEADFIELDS. It contains all fields that simultaneously can depend on the customer and the material. The plant is a prominent example. The plant is therefore empty in CT_KOMK and filled in CT_KOMP_HEADFIELDS instead for each customer/material combination. So, if you have added fields to KOMP or KOMK whose content also depends on header data, you need to add these fields to KOMP_HEAD_FIELDS.

The BAdI interface has a second method called ADAPT_HEAD_ITEM_DATA. You can add fields to the already mentioned source Additional Data here. It is useful if you have fields you want to use in DB aliases because ADAPT_HEAD_ITEM_DATA is processed before the database selects.

Similar to user exits and formulas, you can request the determination of additional fields from KOMK and KOMP for your implementation of method `ADAPT_KOMK_KOMP_DATA`. To do so, go to IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Business Add-Ins (BAdIs) • Define Fields used in BAdI: Price Inquiry Preparation and enter the required fields (see Figure 3.34).


Edit Fields Used in BAdI: Price Inquiry Preparation

ZCL_NPL_ZZPROMO [1]

Doc.structure	Document field	Stat...	Description
KOMP	PRODH1		Selected (added manually)

Figure 3.34 Publishing Fields Needed in the Implementation of Method ADAPT_KOMK_KOMP_DATA

The activity works very similarly to Edit Fields used in Pricing Enhancements; however, there is no code scan. The system will fill the fields requested here, even if the price determination does not need them.

As you can see, the setup of the new price list function requires a certain amount of effort. However, you are usually rewarded with significantly shorter run times.

Check Reports

Currently, two reports exist that allow some control of the settings:

- SD_PRC_MULTI_ITEM_CHECK
  Examines the new bundled access to condition tables, but not the data determination for the KOMK and KOMP fields and is therefore of limited use to the end user. It may be helpful in isolating the cause if the result of the price list is not as expected. It has a very detailed protocol that makes it easy to find inconsistent conditions.
- SDPIQAPICOMPARE
  Compares the price list result with that from an order simulation or the traditional net price list run. The protocol is not as detailed as in SD_PRC_MULTI_ITEM_CHECK.

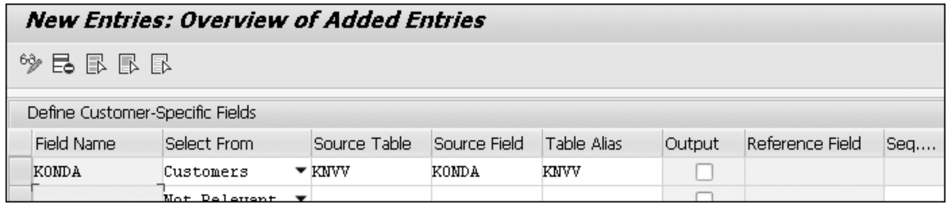
3.4.5 Enhancements of the Selection Screen

You can extend the selection criteria and the result fields of the price list. Let's assume that we want to use the customer price group KONDA as additional selection criteria. To enhance the selection screen, we will have to enhance the price list report SDPIQPRICELIST using enhancement spot `ES_PIQ_SDPRICELIST`. In enhancement point `ES_PIQ_SDPRICELIST_EXT1`, we add the field to the screen. We use enhancement point `ES_PIQ_SDPRICELIST_EXT7` to add a field label and enhancement point `ES_PIQ_SDPRICELIST_EXT6` to transfer our selection option to the program using ABAP macro `transfer_select_option`. Listing 3.1 shows the necessary coding.

```
REPORT sdpiqpricelist.
...
ENHANCEMENT-POINT ES_PIQ_SDPRICELIST_EXT1 SPOTS
                            ES_PIQ_SDPRICELIST STATIC .
*$*$-Start: ES_PIQ_SDPRICELIST_EXT1--------------------
ENHANCEMENT 4  ZZ_NPLTEST.    "active version
 Select-OPTIONS: konda for knvv-konda.
ENDENHANCEMENT.
*$*$-End:   ES_PIQ_SDPRICELIST_EXT1--------------------
...
Customer extension: change of selection field texts
ENHANCEMENT-POINT ES_PIQ_SDPRICELIST_EXT7
                             SPOTS ES_PIQ_SDPRICELIST .
*$*$-Start: ES_PIQ_SDPRICELIST_EXT7--------------------
ENHANCEMENT 3  ZZ_NPLTEST.    "active version
  loop at screen.
    if screen-name = '%_KONDA_%_APP_%-TEXT'.
      %_KONDA_%_APP_%-TEXT = 'Customer Price Group'.
      exit.
    endif.
  endloop.
ENDENHANCEMENT.
*$*$-End:   ES_PIQ_SDPRICELIST_EXT7--------------------
...
transfer the search criteria of customer extension fields to price list
 API
ENHANCEMENT-POINT ES_PIQ_SDPRICELIST_EXT6 SPOTS ES_PIQ_SDPRICELIST .
*$*$-Start: ES_PIQ_SDPRICELIST_EXT6--------------------
ENHANCEMENT 2  ZZ_NPLTEST.    "active version
  transfer-select_option 'KNVV' 'KONDA' konda.
ENDENHANCEMENT.
*$*$-End:   ES_PIQ_SDPRICELIST_EXT6--------------------
```

Listing 3.1 Adding the Customer Price Group KONDA as Selection Criteria

Finally we make an entry in IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Selection and Results • Define Fields because we need to tell the system from which master data table the KONDA value can be derived and how it is connected to the customer or material data (see Figure 3.35).



New Entries: Overview of Added Entries

Define Customer-Specific Fields

Field Name	Select From	Source Table	Source Field	Table Alias	Output	Reference Field	Seq....
KONDA	Customers ▾	KNVV	KONDA	KNVV	□		
	Not Relevant ▾				□		

Figure 3.35 Inform the Price List That Field KONDA Can Be Selected from Table KNVV with Alias KNVV

Similar to the data determination for the pricing communication structure, the information for the generic selection of customers and materials is stored in the form of metadata. The table entry in Figure 3.35 tells the system that, while selecting the customer, it can find the relevant customer group information in field KONDA of table `KNVV`. The table alias `KNVV` is the DB alias you saw in Figure 3.27.

The KONDA field is now available as selection criteria on the selection screen (see Figure 3.37) of the price list.

Additional Fields for Selection and Output

The price list report uses tables `KNVV` and `MVKE` to retrieve all necessary data for selection and output. You can only add fields from related tables such as KNA1 or MARA. The program creates a generic `select` statement joining all necessary tables during runtime. BAdIs exist for more complex cases.

In a very similar way, you can extend the selection screen by your own input parameter. Let's assume you want to provide the price list currency. In this case, you add the code in Listing 3.2.

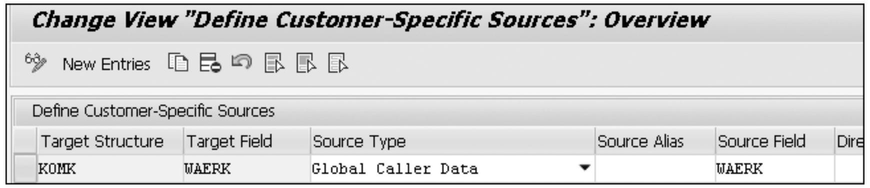
```
REPORT sdpiqpricelist.
...
* Customer extension: further criteria in control block
ENHANCEMENT-POINT ES_PIQ_SDPRICELIST_EXT3 SPOTS
                             ES_PIQ_SDPRICELIST STATIC .
*$*$-Start: ES_PIQ_SDPRICELIST_EXT3--------------------
ENHANCEMENT 8  ZZ_NPLTEST.    "active version
PARAMETERS Waerk like komk-waerk OBLIGATORY.
ENDENHANCEMENT.
*$*$-End:   ES_PIQ_SDPRICELIST_EXT3--------------------
SELECTION-SCREEN END   OF BLOCK control.
...
Customer extension: change of selection field texts
```

```
ENHANCEMENT-POINT ES_PIQ_SDPRICELIST_EXT7 SPOTS ES_PIQ_SDPRICELIST .
*$*$-Start: ES_PIQ_SDPRICELIST_EXT7--------------------
ENHANCEMENT 3  ZZ_NPLTEST.    "active version
  loop at screen.
    if screen-name = '%_WAERK_%_APP_%-TEXT'.
      %_WAERK_%_APP_%-TEXT = 'Price List Currency'.
      exit.
    endif.
  endloop.
ENDENHANCEMENT.
*$*$-End:   ES_PIQ_SDPRICELIST_EXT7-------------------
…
transfer the search criteria of customer extension fields to price list
 API
ENHANCEMENT-POINT ES_PIQ_SDPRICELIST_EXT6 SPOTS ES_PIQ_SDPRICELIST .
*$*$-Start: ES_PIQ_SDPRICELIST_EXT6--------------------
ENHANCEMENT 2  ZZ_NPLTEST.    "active version
  add_name_value 'WAERK' WAERK 'KOMK'.
ENDENHANCEMENT.
*$*$-End:   ES_PIQ_SDPRICELIST_EXT6--------------------
```

Listing 3.2 Adding the Document Currency as Control Parameter

To inform the price list that, from now on, it has to read the currency from the caller data and not from the customer master data, you add the entry shown in Figure 3.36 in IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Price Inquiry • Sources for Communication Structures.

You might have noticed the addition of `OBLIGATORY` in Listing 3.2. This is necessary because with the entry from Figure 3.36, the currency is no longer read from the customer master. Therefore, you must require that the user supply a currency on the selection screen. Alternatively, we could fill the currency in method `ADAPT_KOMK_KOMP_DATA` of BAdI `BADI_PIQ_PREPARE` in case it has not been set by the user.



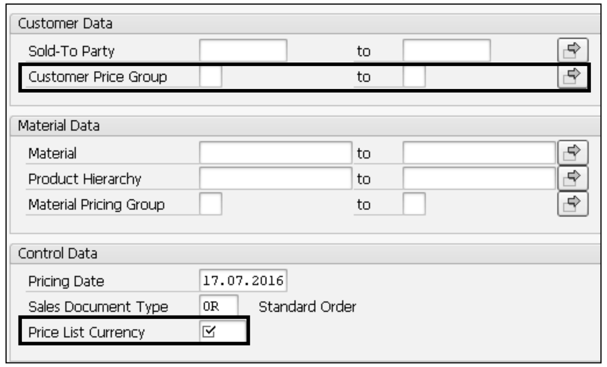
Change View "Define Customer-Specific Sources": Overview

Define Customer-Specific Sources

Target Structure	Target Field	Source Type	Source Alias	Source Field	Dire
KOMK	WAERK	Global Caller Data ▾		WAERK	

Figure 3.36 Retrieve the Currency from the Caller Data

You can see the result of these enhancements in Figure 3.37.



Customer Data			
Sold-To Party		to	
Customer Price Group		to	

Material Data			
Material		to	
Product Hierarchy		to	
Material Pricing Group		to	

Control Data		
Pricing Date	17.07.2016	
Sales Document Type	OR	Standard Order
Price List Currency	☑	

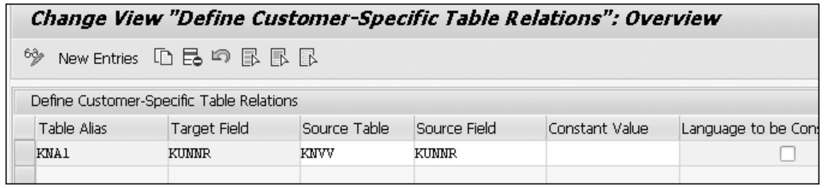
Figure 3.37 Selection Screen Now Enhanced with Customer Price Group and Currency

In addition to the enhancement spot `ES_PIQ_SDPRICELIST`, SAP provides the BAdI `BADI_PIQ_SDPRICELIST` to influence the selection of the price list. Here, the methods `GET_CUSTOMER_DATA` and `GET_MATERIAL_DATA` are particularly worth mentioning. They allow you to add or remove customers and materials from the price list processing. Method `PREPARE_PRICING` permits final changes to global, control, head, and item data before the underlying price inquiry API is called.

3.4.6 Enhancements of the Price List Result

You can also adjust the output to your needs. Suppose that in addition to the customer number, we also want to display the customer's name. We know that the customer's name is stored in the field NAME1 of the customer master table `KNA1`. Selecting the path IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Selection and Results • Define Table Mapping, we find that an access to table `KNA1` with the customer number is already available as the DB alias `KNA1`. Since only the tables `KNVV` and `MVKE` (sales area data for customer and material) are used for the selection and the output of the price list, we need to tell the system how `KNA1` relates to `KNVV`. The system can then select the data with one access by joining the two tables. We choose the customer section using IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Selection and Results • Define Table Mapping and specify that the customer number KUNNR of `KNVV` must be equal to the customer number KUNNR of `KNA1` (see Figure 3.38).

Next, we have to define that the field NAME1 is retrieved from `KNA1`. We go to Customizing activity IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Selection and Results • Define Fields and enter the line shown in Figure 3.39 in the customer section.

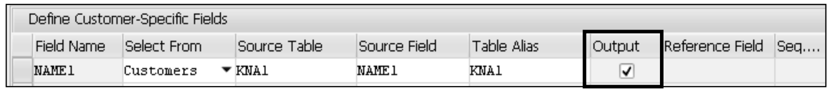


Change View "Define Customer-Specific Table Relations": Overview

Define Customer-Specific Table Relations

Table Alias	Target Field	Source Table	Source Field	Constant Value	Language to be Con
KNA1	KUNNR	KNVV	KUNNR		☐

Figure 3.38 Specifying the Relation between Two Tables



Define Customer-Specific Fields

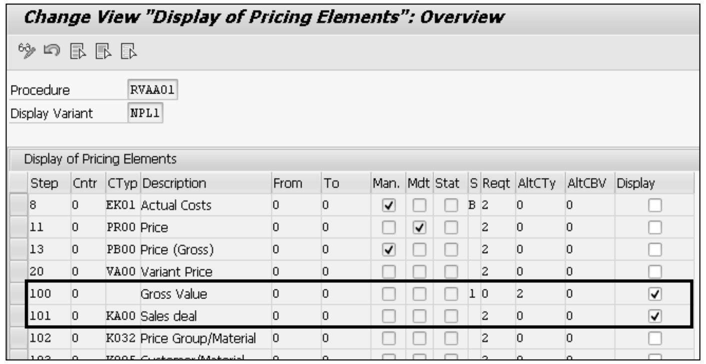
Field Name	Select From	Source Table	Source Field	Table Alias	Output	Reference Field	Seq....
NAME1	Customers	KNA1	NAME1	KNA1	☑		

Figure 3.39 Selecting the Customer Name for Display in the Price List Result

Now the price list will read field NAME1 from table `KNA1` with the key field KUNNR being the same as in `KNVV`. We flag the checkbox Output to add the field to the output data.

In addition, we want to display a particular discount in the price list. As discussed earlier in the chapter, we need to create a display variant to make the selected lines of the pricing result available.

We create a variant and select the rows in the pricing procedure in IMG Customizing activity IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Selection and Results • Define Pricing Elements Display (see Figure 3.40).



Change View "Display of Pricing Elements": Overview

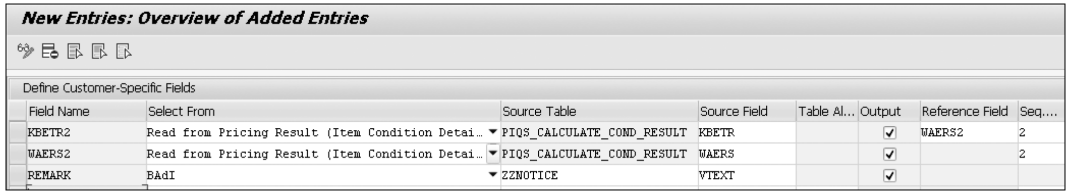
Procedure RVAA01
Display Variant NPL1

Display of Pricing Elements

Step	Cntr	CTyp	Description	From	To	Man.	Mdt	Stat	S	Reqt	AltCTy	AltCBV	Display
8	0	EK01	Actual Costs	0	0	☑	☐	☐	B	2	0	0	☐
11	0	PR00	Price	0	0	☐	☑	☐		2	0	0	☐
13	0	PB00	Price (Gross)	0	0	☑	☐	☐		2	0	0	☐
20	0	VA00	Variant Price	0	0	☐	☐	☐		2	0	0	☐
100	0		Gross Value	0	0	☐	☐	☐	1	0	2	0	☑
101	0	KA00	Sales deal	0	0	☐	☐	☐		2	0	0	☑
102	0	K032	Price Group/Material	0	0	☐	☐	☐		2	0	0	☐
103	0	K005	Customer/Material	0	0	☐	☐	☐		2	0	0	☐

Figure 3.40 Display Variant: Data from the Gross Price Subtotal Line and the Condition Line KA00 Are Provided for the Result List

Similar to the NAME1 fields, we need entries in the customer-specific fields list to make the pricing element details available in the output. First, we have to realize what fields we need. In our case, we want the condition amount KBETR and its currency WAERS from the KA00 condition that is the second price element in our display variant (see Figure 3.40). Therefore, we add the lines shown in Figure 3.41. Price elements are always read from the price result (item condition details) and the corresponding source table is `PIQS_CALCULATE_COND_RESULT` (the exception to this rule is the description of a price element or subtotal line, which is stored in `PIQS_CALCULATE_STEP_DESCR`). Since we need the second line of the variant, we add the Sequence Number 2. To ensure the correct formatting of the amount, we also enter a reference field.



New Entries: Overview of Added Entries

Define Customer-Specific Fields

Field Name	Select From	Source Table	Source Field	Table Al...	Output	Reference Field	Seq....
KBETR2	Read from Pricing Result (Item Condition Detai…	PIQS_CALCULATE_COND_RESULT	KBETR		☑	WAERS2	2
WAERS2	Read from Pricing Result (Item Condition Detai…	PIQS_CALCULATE_COND_RESULT	WAERS		☑		2
REMARK	BAdI	ZZNOTICE	VTEXT		☑		

Figure 3.41 Adding Fields to the Output

Last but not least, we want to add a text field where we can write a short notice. For more complex field determinations, SAP provides the method `PROCESS_RESULTS` in BAdI `BADI_PIQ_SDPRICELIST` where you can fill your output fields dependent on other result data. We add the third line from Figure 3.41, indicating that we will fill the fields in the BAdI. Nevertheless, we have to supply a source table and field to define the technical properties of our field.

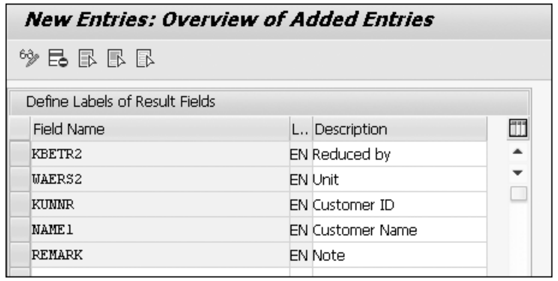
We decide to add text if the discount exceeds a certain limit (see Listing 3.3).

```
METHOD if_badi_piq_sdpricelist~process_result.
  FIELD-SYMBOLS: <ft_result>   TYPE STANDARD TABLE,
                 <fs_result>   TYPE any,
                 <fv_kbetr>    TYPE kbetr,
                 <fv_remark>    TYPE text30.
  DATA: lv_limit TYPE kbetr VALUE '-25.00'.
  DATA: comp1 TYPE komp_type VALUE 'KBETR2',
        comp2 TYPE komp_type VALUE 'REMARK'.
  ASSIGN cr_result->* TO <ft_result>.

  LOOP AT <ft_result> ASSIGNING <fs_result>.
    ASSIGN COMPONENT comp1 OF STRUCTURE <fs_result> TO
                                        <fv_kbetr>.
```

```
      CHECK <fv_kbetr> IS ASSIGNED.
      IF <fv_kbetr> LE lv_limit.
        ASSIGN COMPONENT comp2 OF STRUCTURE <fs_result>
                                    TO <fv_remark>.
        CHECK <fv_remark> IS ASSIGNED.
        <fv_remark> = 'greatly reduced'.
      ENDIF.
    ENDLOOP.

  ENDMETHOD.
```

Listing 3.3 Adding Text if the Discount Exceeds 25%

Finally, we want to adjust the labels of our new columns. To do so, we go to IMG • Sales and Distribution • Basic Functions • Pricing • Price Lists • Selection and Results • Define Labels of Result Fields and enter suitable descriptions (see Figure 3.42). We can define the labels in general or depending on the particular pricing procedure. However, be aware that procedure-related labels do not appear in the Display List option. This output option can combine several customers in one list and there may be more than one pricing procedure involved. The column label needs to be unique, however.



New Entries: Overview of Added Entries

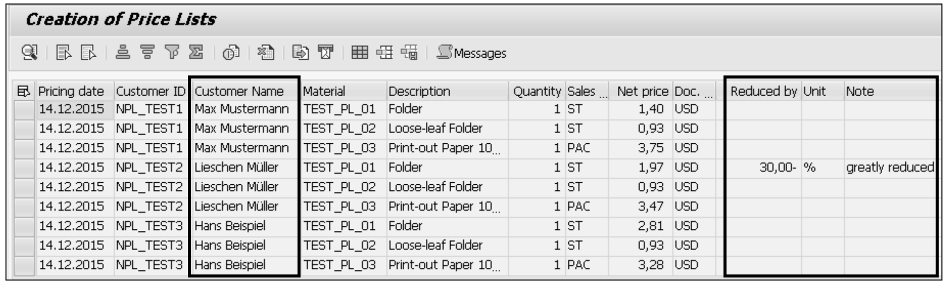
Define Labels of Result Fields

Field Name	L..	Description
KBETR2	EN	Reduced by
WAERS2	EN	Unit
KUNNR	EN	Customer ID
NAME1	EN	Customer Name
REMARK	EN	Note

Figure 3.42 Adjusting the Column Header of the Price List Result

After adjusting the layout of the price list report, the final list will look something like the one shown in Figure 3.43 (Customer Name, Reduced by, and Note have been added).

We do not want to finish this section without pointing out another BAdI. The method `PROCESS_OUTPUT` of BAdI `BADI_PIQ_SDPRICELIST_OUTPUT` allows you to program your own output options. You can, for example, decide to store the price list result in a database table for analytical purposes.



Creation of Price Lists

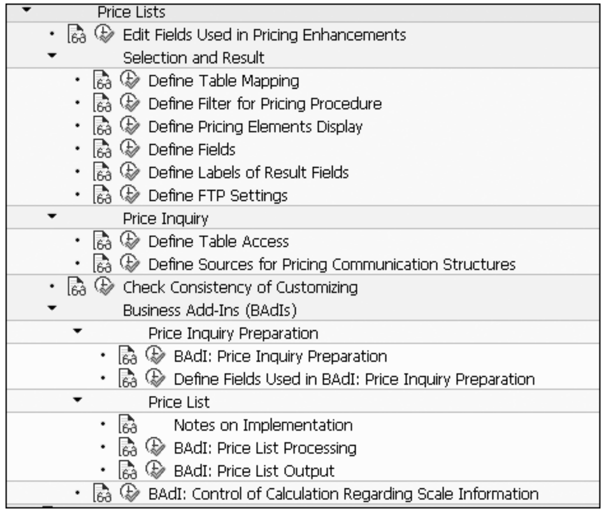
Pricing date	Customer ID	Customer Name	Material	Description	Quantity	Sales ...	Net price	Doc. ...	Reduced by	Unit	Note
14.12.2015	NPL_TEST1	Max Mustermann	TEST_PL_01	Folder	1	ST	1,40	USD			
14.12.2015	NPL_TEST1	Max Mustermann	TEST_PL_02	Loose-leaf Folder	1	ST	0,93	USD			
14.12.2015	NPL_TEST1	Max Mustermann	TEST_PL_03	Print-out Paper 10...	1	PAC	3,75	USD			
14.12.2015	NPL_TEST2	Lieschen Müller	TEST_PL_01	Folder	1	ST	1,97	USD	30,00-	%	greatly reduced
14.12.2015	NPL_TEST2	Lieschen Müller	TEST_PL_02	Loose-leaf Folder	1	ST	0,93	USD			
14.12.2015	NPL_TEST2	Lieschen Müller	TEST_PL_03	Print-out Paper 10...	1	PAC	3,47	USD			
14.12.2015	NPL_TEST3	Hans Beispiel	TEST_PL_01	Folder	1	ST	2,81	USD			
14.12.2015	NPL_TEST3	Hans Beispiel	TEST_PL_02	Loose-leaf Folder	1	ST	0,93	USD			
14.12.2015	NPL_TEST3	Hans Beispiel	TEST_PL_03	Print-out Paper 10...	1	PAC	3,28	USD			

Figure 3.43 The Final ALV Output of Price List Example

SAP Note 1949280 provides some detailed descriptions if you want to engage more deeply in selection and output enhancements.

3.4.7 Customizing Overview

This section serves as a recap and brief overview of the Customizing and Enhancement options of the performance-optimized price list. Figure 3.44 shows the relevant part of the IMG.



- Price Lists
  - Edit Fields Used in Pricing Enhancements
  - Selection and Result
    - Define Table Mapping
    - Define Filter for Pricing Procedure
    - Define Pricing Elements Display
    - Define Fields
    - Define Labels of Result Fields
    - Define FTP Settings
  - Price Inquiry
    - Define Table Access
    - Define Sources for Pricing Communication Structures
  - Check Consistency of Customizing
  - Business Add-Ins (BAdIs)
    - Price Inquiry Preparation
      - BAdI: Price Inquiry Preparation
      - Define Fields Used in BAdI: Price Inquiry Preparation
    - Price List
      - Notes on Implementation
      - BAdI: Price List Processing
      - BAdI: Price List Output
    - BAdI: Control of Calculation Regarding Scale Information

Figure 3.44 IMG Activities for the Performance-Optimized Price List

It helps to understand the structuring, if you know that the price list technically has two layers. The outer one, the *price list layer*, processes the data from the selection screen and builds up the caller data we have seen before. It calls the inner layer, the *price inquiry layer*, and processes the result of the price calculation, for example, by sending the lists to the customers via e-mail. The inner layer is responsible for taking the caller data, deriving from it all necessary information for pricing, performing the call to pricing, and passing the result back to the caller. The price inquiry layer can be called stand-alone and distinguishes its callers by their caller IDs. The price list has the PL caller ID.

The most important activities in the Customizing are as follows:

- Edit Fields Used in Pricing Enhancements
  In this activity, you register fields you need in formulas, requirements, and user exits. With this information, the price inquiry will identify the relevant fields for the price calculation.
- Define Table Mapping
  Either display the standard table aliases and relations, or register your own tables for database `select`s. You need to define how they relate to `KNVV` or `MVKE`. `KNVV` and `MVKE` are the core tables for the selection of customers and products for the price list processing. The defined relation allows a `Join` statement to select all necessary data.
- Define Filter for Pricing Procedure
  In a filter variant, you select lines in a pricing procedure to be skipped during the pricing call. This reduces the runtime of the price list.
- Define Pricing Elements Display
  Lines selected in a display variant are available for output in the price list processing.
- Define Fields
  Display standard fields or register additional fields for selection or output.
- Define Labels of Result Fields
  Adjust the column labels in the output list.
- Define FTP Settings
  Register the server address, user, and password for storing the price list result on an FTP server.
- Define Table Access
  Display the DB aliases of standard table accesses or define new ones. DB aliases define how to access a database table.

- DEFINE SOURCES FOR PRICING COMMUNICATION STRUCTURE
  Display the sources of standard fields or define how your own fields can be determined.
- CHECK CONSISTENCY OF CUSTOMIZING
  Check the consistency of the Customizing.
- BADI: PRICE INQUIRY PREPARATIONS
  BAdI to fill fields in the pricing communication structures.
- DEFINE FIELDS USED IN BADI
  Price Inquiry Preparation: Register fields you need in your BAdI implementation. The system will then provide the necessary information if possible.
- BADI: PRICE LIST PROCESSING
  Allows influencing parameters for the price inquiry call. Allows influencing the result.
- BADI: PRICE LIST OUTPUT
  You can program your own output options here.
- BADI: CONTROL OF CALCULATION REGARDING SCALE INFORMATION
  Influence the standard algorithm and how the relevant quantities are calculated.

3.4.8 Important Programs and Function Modules

Following are the most important programs and function modules of the performance-optimized price list:

- **SDPIQPRICELIST**
  This is the standard report for the creation of price lists. It has a dedicated selection screen allowing you to select customers and products and to choose from various options. You can enhance the report's screen by implementing enhancement spot ES_PIQ_SDPRICELIST.
- **API_PIQ_SDPRICELIST**
  You can use this function module if you want to access the price list functionality from within another program. The interface corresponds more or less to the selection screen of the price list. SAP Note 1949280 provides a detailed description.

- **BAPI_PIQ_SDPRICELIST**
  With BAPI_PIQ_SDPRICELIST, you can call the price list functionality remotely. It is very similar to the function module API_PIQ_SDPRICELIST but allows you to disable the consistency check on the input table for performance reasons. In addition, SAP note 1949280 gives a more detailed description of BAPI_PIQ_SDPRICELIST.
- **PIQ_CALCULATE**
  PIQ_CALCULATE is an RFC-enabled function module and can be useful if you need to calculate prices outside a document simulation with good performance. It requires some control data and a list of customers and materials. It reads metadata from Customizing to optimize the database accesses. Different callers can register different caller IDs in table PIQC_API_CALLER or use the standard ID STD. When using your own caller ID, you need to define the sources for the pricing communication structure for your fields and your new ID in table PIQC_PREP_FLD_C. Unlike the price list APIs, the price inquiry can trigger a group processing for items belonging together.

  If you're using PIQ_CALCULATE, keep in mind that it does not create any document simulation. Hence, not all functionality related to pricing is available. Examples are free goods, product substitution, down payment functionality, EAN numbers, batch determination, etc.

3.5 Summary

With the ability to configure pricing reports, you can ensure, for example, as department supervisor, that pricing reports are provided for your departments that are adapted to their specific information needs. Pricing reports can be created without programming by a kind of list generator in Customizing for pricing.

In addition to the traditional pricing reports, which have also been introduced as a filter medium in the condition maintenance via area menus, we have—starting with SAP ERP 6.0 (but subject to the activation of the business function SD_01)—the worklist for price data from the Internal Sales Representative role as a new development.

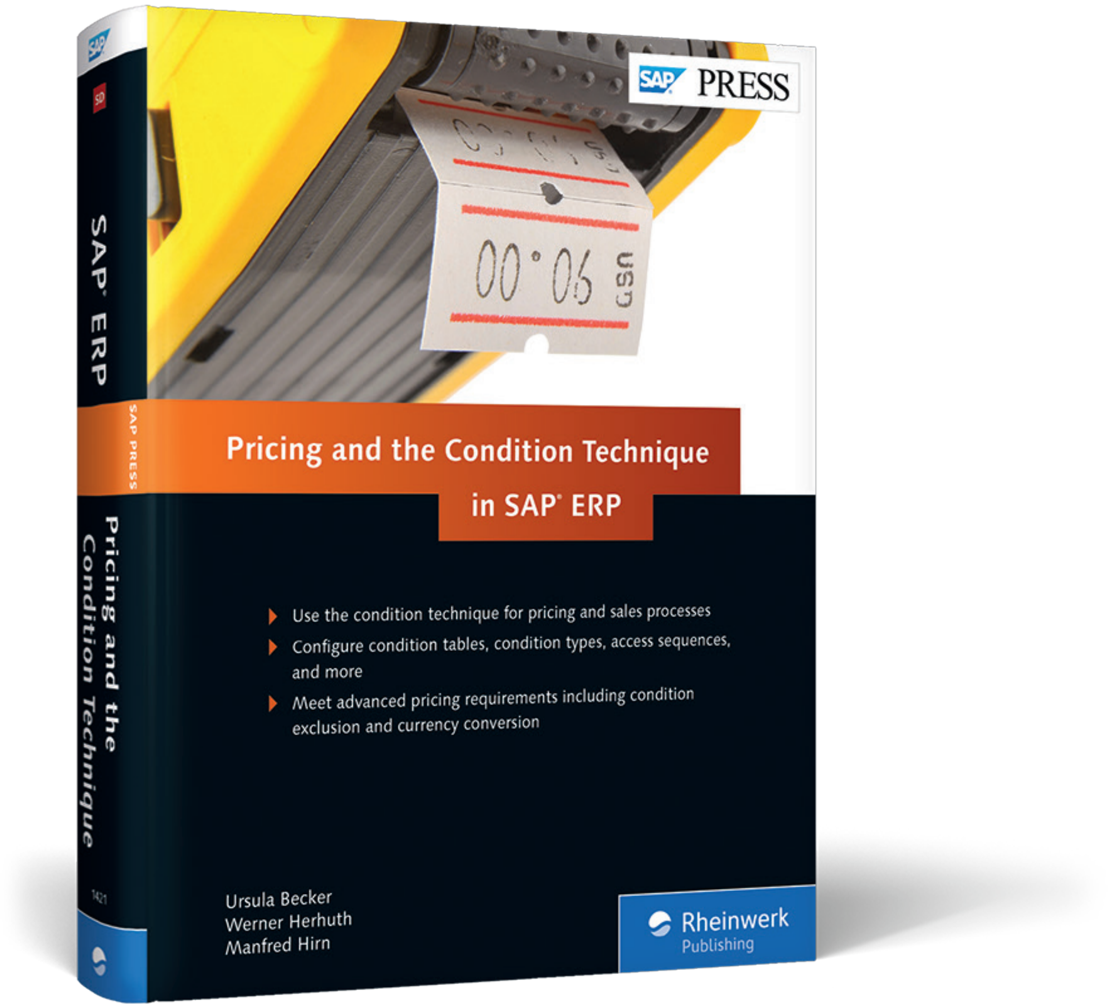
With the net price list or its successor, the performance-optimized price list, you can give your customers individual price information for a large number of products.

This chapter marks the end of the first part of this book about the condition technique. We now move on to the second and central part of this book, standard pricing configuration.

Index

SAP PRESS First-hand knowledge.

Rheinwerk Publishing


SAP PRESS
Pricing and the Condition Technique in SAP® ERP
Use the condition technique for pricing and sales processes
Configure condition tables, condition types, access sequences, and more
Meet advanced pricing requirements including condition exclusion and currency conversion
Ursula Becker
Werner Herhuth
Manfred Hirn
Rheinwerk Publishing
SAP ERP
Pricing and the Condition Technique

Ursula Becker, Werner Herhuth, Manfred Hirn

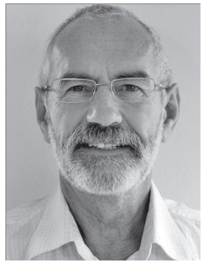
Pricing and the Condition Technique in SAP ERP

665 Pages, 2017, $89.95
ISBN 978-1-4932-1421-1

www.sap-press.com/4173

Dr. Ursula Becker has been the development architect responsible for the pricing functionality in SAP ERP since 2009. She studied physics and astronomy at the University of Heidelberg and joined SAP in 1998, after completing her doctorate and two years of research. In the following years she worked in support before moving on to development—with a focus on pricing, billing ,and rebate processing. She has also worked in the areas of SAP Customer Relationship Management (SAP CRM) and SAP Business ByDesign.

Werner Herhuth is a certified consultant in the area of order fulfillment (in SAP ERP) and the author of several SAP courses. He studied mathematics and business administration with specialization in business informatics at the University of Mannheim. He worked for 13 years as a systems analyst and head of application programming in the engineering industry before joining SAP in 1996.



Manfred Hirn was responsible for the development of the condition technique, as well as the billing and pricing functionalities which he helped program, in SAP R/3. He studied mathematics with a minor in business administration at the University of Würzburg. In his first job at a brewery, he was involved in the development and introduction of an order and trip management system. In 1984, he joined the SAP development team and supervised the SAP R/2 billing program. Later he worked as a development manager in the areas of sales, pricing, and billing.