BaanErp5.0b Manufacturing
Product Configuration (PCF)

Functional Release Information
P3096A US
Document information

Document

Document code : P3096A US
Document group : Product Information
Document title : Product Configuration (PCF)
Application/Package : BaanErp5.0b Manufacturing
Edition : A
Date : May-1999

© Copyright 1999 Baan Development B.V. All rights reserved

The information in this document is subject to change without notice. No part of this document may be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Baan Development B.V.

Baan Development B.V. assumes no liability for any damages incurred, directly or indirectly, from any errors, omissions or discrepancies between the software and the information contained in this document.
Table of contents

About this document

1. The Product Configuration (PCF) module in perspective
   1.1 The Product Configuration module in BaanERP 5.0b
   1.2 The business objects of the PCF module
   1.3 Master Data and prerequisites

2. The Product Features business object
   2.1 Description of Product Features
   2.2 Mandatory sessions
      2.2.1 Product Features (tipcf0150m000)
      2.2.2 Product Features by Generic Item (tipcf1101m000)

3. The Product Constraints business object
   3.1 Description of Product Constraints
   3.2 Mandatory sessions
      3.2.1 Constraints by Generic Item (tipcf2110m000)
      3.2.2 Compile Constraints by Generic Item (tipcf2201m000)
      3.3 Other sessions
         3.3.1 Constraint IDs by Generic Item (tipcf2101m000)
         3.3.2 Constraint Validation Messages (tipcf2120m000)

4. The Generic Engineering Data business object
   4.1 Description of Generic Engineering Data
   4.2 Mandatory session
      4.2.1 Settings for Generic Item-Data Generation (tipcf3101m000)
      4.2.2 Generic BOMs (tipcf3510m000)
      4.2.3 Generic Routing (tipcf3520m000)
   4.3 Other sessions
      4.3.1 Where-Used Components (tipcf3515m000)

5. The Generic Price List business object
   5.1 Description of Generic Price List
   5.2 Mandatory sessions
      5.2.1 Generic Price Lists (tipcf4101m000)
   5.3 Other sessions
      5.3.1 Price List Matrix Codes (tipcf4110m000)
      5.3.2 Price List Matrices (tipcf4120m000)

6. The Product Variant Configuration business object
   6.1 Description of Product Variant Configuration
   6.2 Mandatory sessions
      6.2.1 Product Variants (tipcf5501m000)
      6.2.2 Generate Budget Structure for Sales Quotations (tdsls1201m100)
      6.2.3 Generate Project Structure for Sales Orders (tdsls4244m000)
      6.2.4 Generate Product Variant Structure for Project (tipcs2220m000)
      6.2.5 Element Budget (Material Lines) (tppetc1510m000)
6.2.6  Activity Budget (Material Lines) (tpptc2510m000)  6-3
6.3  Other sessions  6-3
6.3.1  Product Variant Option Sets (tipcf5510m000)  6-3
6.3.2  Product Variant Options (tipcf5520m000)  6-3
6.3.3  Product Variant Sales Price Structure (tipcf5530m000)  6-3
6.3.4  Calculate Product Variant Sales Price Structure (tipcf55230m000)  6-3
6.3.5  Archive Product Variants (tipcf55290m000)  6-3
6.3.6  Print Product Variant Statistics (tipcf5410m000)  6-4

7.  The procedure to define a product model  7-1
7.1  The procedure to define a product model and its result  7-1
7.2  The procedure to define a product model  7-2
7.2.1  Model on paper  7-3
7.2.2  The steps of the procedure  7-4
7.3  System test of the product model  7-5
7.3.1  System test of the product features  7-5
7.3.2  System test of the product structure  7-6
7.3.3  System test of the price list  7-7

8.  The procedure to use a product model in the Sales Order Management module  8-1
8.1  The procedure to use a product model in the SLS module and its result  8-1
8.2  The procedure to use a product model in a sales quotation  8-2
8.3  The procedure to use a product model in a sales order  8-3

Appendix A  Examples of constraints  1
About this document

Structure
This document supports the BaanERP 5.0b Product Configuration (PCF) module that is part of the Manufacturing package. Part One describes the Product Configuration (PCF) module as a tool; Part Two describes how the tool can be used.

Part One
Chapter 1 places the module in perspective of BaanERP 5.0b, describes the module and explains the relationships with other modules. It also lists all business objects and describes the flow between them. Finally it lists the prerequisites and the master data needed to run this module.

Chapter 2 through 6 describe the business objects. Each chapter explains the flow of the mandatory steps within a business object. An explanation of the optional steps concludes each chapter.

Part Two
The chapters in this part describe the functional procedure chapters. Each chapter provides a detailed explanation of the flow of a functional procedure.

On-line Help
For detailed descriptions of this module there is comprehensive on-line help available in the system.
1. The Product Configuration (PCF) module in perspective

This chapter describes the main functions and objectives of the Product Configuration module. Information is given about the module’s function within BaanERP 5.0b and how the module is related to other modules. Additionally, this chapter presents the module’s business objects, then closes with the global set-up requirements for the module.

1.1 The Product Configuration module in BaanERP 5.0b

The competitive power of a company is determined more and more by the speed with which it can meet customer requirements. Today’s business should be capable of delivering a customer specific product within the delivery time of a standard product.

In a traditional production control system, the product structure generally consists of:

- item data, such as delivery time and cost price
- data relating to the structure of items, such as bills of material
- data about operations, such as routings

This system may be adequate for companies producing only one or just a few products. However, if a large number of variants of the finished product are produced, they are usually only assembled or manufactured when the customer's order has been received. In such cases, the traditional information system may run into problems with respect to the quantity, complexity, and manageability of the product data and the need for timely availability of the information.

Almost any company that is assembling to order deals with product variants. In that case, it is impossible to define the product structure for all the versions of all finished products in advance. The answer to this problem is configuration management. This can be translated into a well thought out, modular, product design with proper validation and decision support functions provided by the information system to enhance the level of logistic control.

In the Product Configuration (PCF) module, a product model is created, in which all the features of the product model are defined. By selecting the options of the features, the customer defines the desired product variant. The translation of the customer requirements into the product structure of the variant is controlled by a set of decision rules (constraints). These constraints indicate which components and operations will or will not be used in a specific version.

Item data, such as the code and a description of an item, can be generated by the PCF module. A generic price list can be created in which matrices can be used for product features which are dependent on each other for what affects the price. Constraints are used to determine the item data and the sales and purchase price of the generic items.
The product model is used in sales orders and quotations to translate the customer requirements into a product variant. A budget is generated for the quotation to create the product structure (bills of material and routing) of the variant. For this purpose, a project is generated for an order. The sales price from a sales quotation is compared to the budget to ensure that it meets sales margin requirements.

A PCS project is used to plan, produce, and control the manufacturing of the product variant. With the statistics in the PCF module, insight can be obtained about the selected options of the product model. This information can be used, after analysis, as input for a new master production schedule for the next period.

Figure 1-1 The PCF module in BaanERP 5.0b

In BaanERP 5.0b, the PCF module is used to create a product model that can be used in a sales quotation, sales order or BaanERP Project to define a product variant, based on the options selected by the customer. This information will be used to generate a sales price.
The structure of the product variant is generated in a budget or a PCS project. The item data for the generic items can be automatically defined.

Figure 1-2 The modules related to the PCF module

The PCF module uses the Item Base Data (IBD), Item Production Data (IPD) and Routing (ROU) modules to define the generic BOM and routing for the product model. If a manufactured item is defined in the generic BOM, the associated bill of material from the BOM module is used as a part of the total product structure.

A product variant and its structure can be generated for a budget and a project in the Project Control System (PCS) module from a product model in PCF. The budget is used to calculate the cost price; the project is used for planning, production, and control of the product variants.

The relationship between the Product Configuration and the Sales Control (SLS) modules provides the ability to generate product variants with a sales price for sales quotations and orders, using a product model.

An item that is controlled by the Master Planning algorithm of Enterprise Planning (EP) can be a generic item as well. In that case, the product model is associated with a master production plan in Enterprise Planning.

In the generic BOM a planning percentage can be entered for items that are controlled by the Material Requirements Planning algorithm of EP. In this case, it is possible to subdivide the demand for a product from a higher level into forecasts for different options within the bill of material. A condition for this situation is, that the product on that higher level is controlled by the Master Planning system.

A customized routing sheet will be defined when a product variant structure is generated for a project in the PCS module. Planned capacity is visible in the Enterprise Planning (EP) module for all the work centers in the routing.

It is also possible to define a configured product as part of the BaanERP Project package.
1.2 The business objects of the PCF module

The business objects belonging to the Product Configuration module are:

- Product Features
- Product Constraints
- Generic Engineering Data
- Generic Price Lists
- Product Variant Configuration

![Diagram of the main flow between the business objects in the PCF module](image)

Product model definition begins with outlining the model on paper. By creating a model in which all possible variants are defined, it becomes easier to enter the bills of material and routing in BaanERP 5.0b. If the model is drawn on paper, the next step is to use the PCF module functionality.

An item of type ‘generic’ is defined that represents all the possible variants of a product. The features of the product model, with potential options, are recorded in the Product Features business object. These features are the basis for a question and answer procedure used to record the product specifications.
By using restrictive rules, called constraints, certain selections can be excluded or made mandatory. Certain questions are asked by BaanERP 5.0b depending on the answers to previous questions. BaanERP 5.0b may provide default answers or, in some cases, directly produce the answer. The Product Constraints business object controls this.

In addition to controlling the set of features, these constraints are also important for the BOM, routing, item data, and the price lists. Constraints ensure that the options selected by the customer are translated into the appropriate product structure with the correct prices and item data.

In the Generic Bills of Material and Routing, the product structure for all the possible variants of a product model is entered. The bills of material use the data from the Item Base Data (IBD) and Item Production Data (IPD) module for the standard items that are a part of the product structure used to assemble the end product. Planning percentages for plan items can be defined in the BOM, if a Master Plan is used at a higher level. A forecast can be created for these items making use of the master plan for the generic item.

The generic routing uses the data from the Routing (ROU) module to define work centers, tasks, and machines. The operations in the routing can be planned by generating a product variant structure for a project. This information is visible in the Enterprise Planning (EP) module.

The Generic Engineering Data business object can generate specific item data for a product variant, such as item code and descriptions. By using this functionality, a customized item is defined in Project Control System (PCS).

A generic price list can be defined in the Generic Price Lists business object. Matrices can be used if options for product features that are dependent on each other influence the purchase or sales price for a product variant.

If the Product Variant Configuration is used in the Sales Control (SLS) module, a sales price based on the data in the Generic Price List business object can be generated.

The purchase price can also be generated; it is used to determine the cost price of a product variant in the PCS module.

The product model can be used in the operational company if the data in all the previously mentioned business objects are filled. Product variants are generated from the options selected by the customer in the Product Variant Configuration business object. The functionality of the PCF module can be used in sales order and quotations in Sales Control (SLS) module. The structure of the variants is generated for a budget or a project in the PCS module.

A product variant can be archived if it is delivered to the customer and the project is closed. By archiving, the statistics in the PCF module are created for the options selected by the customer. This information can be used for a new period in the master production schedule in the EP module.
1.3 Master Data and prerequisites

This module can only be used if the PCF module Implemented is set to yes in the Company Data (tccom0500m000) session.

Work centers and tasks must be defined in the ROU module before generic routings can be entered.

Generic items and standard items must be defined with the General Item Data (tcibd0501m000) and Item Production Data (tiipd0501m000) session before a product model is defined or a generic BOM is created.
2. The Product Features business object

This chapter describes the Product Features business object and gives information about the business object’s functions. In addition, the chapter identifies and explains the mandatory and optional sessions related to the business object. Related parameters are mentioned where applicable.

2.1 Description of Product Features

The product model is used in a question and answer situation to define the product and its total structure. The features that are defined in this business object are used as the questions for the customer. The answers to the questions are used in the PCF module to determine the product structure, the sales price, and the generic item data for the variant. The functionality of this business object is used at the start of the procedure to create a product model. It is also used when a product variant is generated in the operational situation.

2.2 Mandatory sessions

2.2.1 Product Features (tipcf0150m000)

Properties can be used in a question and answer situation to make it possible to describe the product needed by the customer. These product features can use options that are defined in the session Options by Product Feature (tipcf0160m000). One product feature can consist of multiple options. The features defined in this session are product independent. It is possible to zoom from this session to the Options by Product Feature (tipcf0160m000) session.

If there is a lot of communication with external connections and customers in other languages, the features and potential options can be translated into different languages. The PCF in Customer's Language parameter must be set to yes in the PCF Parameters (tipcf0100s000) session.
2.2.2 Product Features by Generic Item (tipcf1101m000)

Product features are used as the questions to create a product model. The questions are associated to a product model by entering the appropriate product features, by generic item, in the correct sequence.

Product features and options are automatically taken by the lower levels in the configuration structure from the higher levels; therefore, it is not necessary to record these features at each level in the configuration structure. Product features can be controlled by date-effectivity to ease the phase in of new product features. Product features become product dependent in this session.
3. The Product Constraints business object

This chapter describes the Product Constraints business object and gives information about the business object’s functions. In addition, the chapter identifies and explains the mandatory and optional sessions related to the business object. Related parameters are mentioned where applicable.

3.1 Description of Product Constraints

A constraint enables every possible decision rule or calculation conceivable, that can be executed during the definition of the variants, to be built. These decision rules can be used in the product model for the product features, generic BOM, routing, price list, and item data. These rules are defined with a constraint editor.

A meta-language specifically devised for this purpose is used to define constraint expressions; it is a subset of BaanERP 5.0b Tools. The functionality of this business object is used during the entire procedure to create a product model.

In the operational situation, the constraints ensure that the customer requirements are translated into a product variant with an associated structure. See Appendix A for examples of constraints.

3.2 Mandatory sessions

![Figure 3-1 Mandatory sessions within Product Constraints](image)

The figure shows the mandatory sessions within the Product Constraints business object.
3.2.1 Constraints by Generic Item (tipcf2110m000)

Constraints provide a flexible way to define a generic product model. These constraints are used as decision rules. Constraints make it possible to indicate under what conditions certain combinations of options are acceptable, mandatory, or not acceptable for product features, bills of material components, operations, item data, and price lists.

In this session, the constraints are defined for a generic item with the constraint editor (either the Baan Editor (ASCII) or graphical multi line editor). In PCF Parameters (tipcf0100s000) session a number of constraint expression variables are predefined. These variables can be furnished with descriptions, or names, only once. After product constraints have been defined, the variables’ names can no longer be changed.

Various variables can be identified in the Product Configurator:

- parameter variables
- local and global variables
- product features

The parameter variables must be recorded in the PCF Parameters (tipcf0100s000) session. These variables are used to formulate the constraints.

Local and global parameters can be created. By adding global, a local variable is converted into a global variable. A local variable is only valid for the specific constraint in which it is used; a global variable can be used in more than one constraint. The variable must be declared at the beginning of the constraint text in each situation where the global variable is used in the constraint.

Product features are also global variables. The product features can be used throughout the entire PCF model. After a product feature receives a particular value, it keeps this value until another value is entered for it.

The function of the Constraints by Generic Item (tipcf2110m000) session is to record constraints for each constraint code. There are three constraint sections that can be used:

- before input
- validation
- parameter substitution

A constraint always uses a constraint section to define the decision rule. The following table provides a description about the different constraint sections.

The possible combinations between constraint sections and the various possibilities for which the constraints are formulated, are shown in the following table.
The Product Constraints business object

Possible combinations of constraint sections

<table>
<thead>
<tr>
<th>constraint section for</th>
<th>before input</th>
<th>validation</th>
<th>parameter substitution</th>
</tr>
</thead>
<tbody>
<tr>
<td>generic BOM</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>generic routing</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>product features</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>generic price list</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>generic item data</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>used for</td>
<td>default input calculations</td>
<td>validations calculations messages</td>
<td>parameter variables</td>
</tr>
</tbody>
</table>

The "before input" option is a constraint section that is focused on providing a default option value for a particular feature. Based on particular selections, BaanERP 5.0b presents a recommended option for a particular characteristic. This option value can be overwritten by another, providing input is allowed.

The "validation" constraint excludes or includes particular option selections; it validates, performs calculations, and can be used to display messages.

The "parameter substitution" constraint ensures that a certain calculated value or a value that is based on selections is entered or exported.

It is possible to define different constraint sections for each constraint code.

3.2.2 Compile Constraints by Generic Item (tipcf2201m000)

The use of the product configurator can be divided in two parts. The first part is product model definition; the second part is the use of this product model in an operational situation. In the development stage, the constraints must be tested to ensure that no errors are present, so that the model can be used in the operational situation.

For this development stage an interpreter version can be used. This version is defined in the PCF Parameters (tipcf0100s000) session. Using this version means that constraints are constantly read in and interpreted. The disadvantage of this version is that the performance of the product configurator is reduced.

That is why a different version is available for the operational situation, the object version. In this version, constraints are directly executed by the product configurator as objects, without needing to repeatedly read in and interpret or compile constraints in run time.

The object version of the product configurator may result in a considerable performance improvement compared to the interpreter version, particularly with more complex constraints.

By using the Compile Constraints by Generic Item (tipcf2201m000) session, an object will be generated for each item. For that reason, it is possible to use the object version.
3.3 Other sessions

3.3.1 Constraint IDs by Generic Item (tipcf2101m000)

Constraints can be defined in different PCF business objects. By creating these constraints from within these business objects, a constraint ID is generated. If a company wants to use identification codes, different from the generated codes, the functionality of this session enables identification code definition.

3.3.2 Constraint Validation Messages (tipcf2120m000)

During the configuration process, a user sometimes needs information in addition to the descriptions of the product features. For example, the information can include: additional information about certain options or information if a mistake is made. This information can be provided by using messages.

If a constraint is entered by using the validation constraint section, messages can be defined. The information can be entered as text by using the constraint expression `message`. The information can also be entered as a code, defined in this session, and used in the constraint for the same constraint expression.

An advantage of using this session for defining the messages is that if changes need to be made to the messages, it is easy to have all messages in one session.
This chapter describes the Generic Engineering Data business object and gives information about the business object’s functions. In addition, the chapter identifies and explains the mandatory and optional sessions related to the business object. Related parameters are mentioned where applicable.

### 4.1 Description of Generic Engineering Data

The Generic Engineering Data business object consists out of the following mandatory sessions:

- **Settings for Generic Item Data Generation**
  Data for generic items can be filled with customer specific data. The process for creating item data is controlled with constraints. The functionality of this business object can be used during the entire procedure to create a product model.

- **Generic Bills of Materials**
  The PCF module uses a generic product model to generate product variants with bills of materials defined by customer specification for the end product. The process for the creation of the bills of materials is controlled with constraints. The generic BOM can serve as the basis for forecast calculation in Enterprise Planning. The functionality of this business object can be used during the entire procedure to create the product model. It is also used in the operational situation, if a product variant is created based on the requirements of the customer.

- **Generic Routing**
  By using a generic product model, the PCF module can generate new product variants with a routing defined by customer specifications for the end product. The process for creating the routing can be controlled with constraints. The data of the generic routing is used for capacity requirements calculation in the Enterprise Planning module. The functionality of this business object can be used during the entire procedure to create the product model. It is also used in the operational situation, if a product variant is created, based on the requirements of the customer.
4.2 Mandatory session

4.2.1 Settings for Generic Item-Data Generation (tipcf3101m000)

The customer’s selections can be used to create item codes and descriptions. For example, material, size, and standard of the customized items and texts for quotations and orders can be filled with situation dependent and fixed data.

The created item data occurs if the structure of a product variant is generated for a budget or a project. With a budget, the cost price of the product variant can be calculated; a project is used to manufacture the product variant.

The Generate Project Structure for Sales Orders (tdsls4244m000) session is used to define the project. If an item code is generated for a project that corresponds with a standard item code with inventory on hand, BaanERP 5.0b includes this item in the sales order and a project is not created.

Some parameters influence the functionality of this session. These parameters are all defined in the Maintain PCF Parameters (tipcf0100m000) session. If an identifying item code must be used, set the Identifying Item Code parameter to yes.

If the item code generated by the product configurator does not uniquely identify a product variant, the product configurator can automatically generate a unique code.

The product configurator adds a sequence number to the generated item code using the sequence number from Item Code from Position parameter in PCF Parameters (tipcf0100m000) session.

This parameter is used to indicate from which position of the item code the sequence number must be generated.

4.2.2 Generic BOMs (tipcf3510m000)

The customer’s selections can be used to generate bills of materials. Constraints control the translation of customer requirements to the components of the bills of materials for product variants. In this session, the generic BOM is defined with all possible components.

A planning percentage can be entered for the components of the BOM that are controlled by the MRP algorithm of EP. This makes it possible to create a forecast of the demand for this item, if the generic item at a higher level uses the Master Planning order system.

If the Multi-Level PCF Choice Structure parameter is set to yes in the PCF Parameters (tipcf0100s000) session, extra generic levels in the product structure can be used for a multi-level selection structure.
A restriction applies to Generic BOM’s. Generic BOM’s can not make use of the new Scrap quantity calculation (to include in MRP) that is introduced for production BOM’s in BaanERP 5.0b.

4.2.3 Generic Routing (tipcf3520m000)

The customer’s selections can be used to generate a routing. Constraints control the translation of customer requirements to a set of operations that must be executed to manufacture the product variant. In this session, the generic routing is defined with all possible operations.

Some remarks have to be made in comparison between Generic Routings and Routings as implemented in BaanERP 5.0b.

The “Task – Workcenter – Machine” relationship is implemented as in a standard Routing. This has two implications:

1. Task - workcenter has to be equal to machine-workcenter, when both are filled.

2. When the field ‘Multiple Work Centers’ of a task has the value ‘Only Default’, only the default work center can be used on that task; if it has the value ‘Specific’, the work centers as defined in the session ‘Work Center/Task Relationships’ (tirou0504m000 / tirou0104s000) for this task, can be used; when it has value ‘All’, all work centers can be used on the task.

Scrap and yield for routings, as new functionality in BaanERP 5.0b, can NOT be used in combination with a Generic Routing in PCF.

4.3 Other sessions

4.3.1 Where-Used Components (tipcf3515m000)

Executing this session will search specified BOMs for which components are used in which BOMs. This allows an easy maintenance of BOMs.
5. **The Generic Price List business object**

This chapter describes the Generic Price List business object and gives information about the business object’s functions. In addition, the chapter identifies and explains the mandatory and optional sessions related to the business object. Related parameters are mentioned where applicable.

### 5.1 Description of Generic Price List

A product variant that is generated from customer specifications can have a detailed sales price based on the selected options. Purchase prices for generic items can also be generated. The purchase price is used to calculate the cost price. Matrices can be defined if there are options for different product features that have mutual relationships that influence the purchase or sales price.

The sales and purchase price calculation is controlled with constraints. The functionality of this business object can be used during the entire procedure to create the product model. However, in most cases, the data is filled after defining the features, generic bills of materials, and routing for the product model. It is also used in the operational situation, if a sales price must be calculated for a product variant.

Keep in mind that the sessions mentioned in this object differ fundamentally from the new pricing module in BaanERP 5.0b where Price Matrices, Price Books and e.g. Discount Schedules can be defined. This Generic Price List object is not directly integrated with the previously mentioned objects.

### 5.2 Mandatory sessions

#### 5.2.1 Generic Price Lists (tipcf4101m000)

The customer’s selections can be used to generate sales and purchase prices. The price lists are defined in this session. Constraints control the translation of customer requirements into the sales price for the product variant and cost prices for generic purchase items.

For this session a restriction applies with regard to standard ERP 5.0b functionality. Multi-currency functionality as implemented in standard BaanERP 5.0b is NOT supported. This would only have applied to price list with the type ‘Purchase Price List’. So a product variant price, prices on generic price lists and prices on price list matrices can only be specified in one currency. This is similar to Baan IVc.
5.3 Other sessions

5.3.1 Price List Matrix Codes (tipcf4110m000)

If there are features with mutual relationships that will influence the price, a matrix can be entered with X and Y-Axis with defined coordinates. The related features and their coordinates are defined in this session.

5.3.2 Price List Matrices (tipcf4120m000)

A value must be defined for every existing relationship between options of different features. This value is used for the calculation of the prices and can be a surcharge or a discount.

Using the value recorded in the Arithmetical Expression field and/or in the Amount Type field in the Generic Price Lists (tipcf4101m000) session, the product generator determines whether the values in the price list matrix must be considered as prices or as price factors.
6. The Product Variant Configuration business object

This chapter describes the Product Variant Configuration business object and gives information about the business object’s functions. In addition, the chapter identifies and explains the mandatory and optional sessions related to the business object. Related parameters are mentioned where applicable.

6.1 Description of Product Variant Configuration

This business object enables the maintenance of product variants, and the configuration of variants for sales quotations, sales orders, budgets, and projects. When a sales order line for a configured item is being entered, BaanERP 5.0b can calculate on-line the lead time for producing the make-to-order parts (final assembly schedule lead time), regarding (non)existing inventory of standard parts. It also calculates the variant’s sales price.

The functionality of this business object is used in the operational situation to generate product variants based on options selected by the customer. It uses the product model that is created in the business objects described in the previous chapters.

6.2 Mandatory sessions

The sessions in this business object all generate product variants. However, they each belong to a separate functional procedure and can thus e.g. be found in the Sales Order Management object. These sessions have in common that they all generate product variants, whether they are used in a sales order, a sales quotation, a BaanERP Project or a budget.

6.2.1 Product Variants (tipcf5501m000)

This session is used to generate a product variant for a generic item. This product variant stores the options selected by the customer. The product variant code makes it possible to generate various variants for a generic item.

In most cases, this session is used automatically by BaanERP 5.0b. This is also true for the other sessions in this business object that generate data. This session can be used manually if a customer wants information about options and prices without any obligation. In the implementation stage of the product model, it is also used manually to execute system tests.

*The following set of sessions all belong to a different business object, however they all make use of PCF functionality.*
6.2.2 Generate Budget Structure for Sales Quotations (tdsls1201m100)

This session generates a product variant structure for a budget starting from a sales quotation, based on the options selected by the customer. The structure is translated into calculation parts, and material and routing sheets.

This step is used to calculate the cost price of the selected product variant. In the sales quotation line, a relationship is made with the budget for which the structure is generated.

6.2.3 Generate Project Structure for Sales Orders (tdsls4244m000)

This session generates a product variant structure for a project starting from a sales order, based on the options selected by the customer. The structure is translated into customized items, customized bills of material, and customized routing. This step is used to plan, produce, and control the manufacturing process of the product variant in the PCS module. In the sales order, a relationship is made with the project for which the structure is generated.

6.2.4 Generate Product Variant Structure for Project (tipcs2220m000)

This session generates a product variant structure for an existing PCS project (note: there is a fundamental difference between a PCS project and a BaanERP Project – project!). The structure is translated in customized items, customized bills of materials, and customized routing.

This session can be used in two ways. The first is to use a standard product variant created by the Product Variants (tipcf5501m000) session to create the structure of an existing project. The second is to access the Product Configurator (tipcf5120m000) session from this session, so that a product variant is generated. The structure of this product variant will be used for an existing project.

6.2.5 Element Budget (Material Lines) (tpptc1510m000)

This session can be used to add a customized item as a material line to an Element Budget of the BaanERP Project package. The Product Configurator (tipcf5120m000) session can be accessed from this session, so that a product variant is generated.
6.2.6 Activity Budget (Material Lines) (tpptc2510m000)

This session can be used to add a customized item as a material line to an Activity Budget of the BaanERP Project package. The Product Configurator (tipcf5120m000) session can be accessed from this session, so that a product variant is generated.

6.3 Other sessions

6.3.1 Product Variant Option Sets (tipcf5510m000)

A generic item structure can consist out of multiple other generic items. If a Product Variant is generated for a generic item that consists out of multiple other generic items, an Option set will be generated per generic item. This option set than contains the selected options per feature.

6.3.2 Product Variant Options (tipcf5520m000)

This session can be used to show the selected options for the fully configured Product Variant. An already generated Product Variant can not be changed anymore from this session.

6.3.3 Product Variant Sales Price Structure (tipcf5530m000)

From this session it's possible to execute a form command to calculate the sales price structure of a configured variant. This calculation can also be executed directly (see 6.3.4 Calculate Product Variant Sales Price Structure)

6.3.4 Calculate Product Variant Sales Price Structure (tipcf5230m000)

This session calculates the sales price for (a selection of) product variants. To calculate the sales price of a product variant, it is necessary that a generic price list is established for the generic main item.

6.3.5 Archive Product Variants (tipcf5290m000)

Data can be transferred to an archive company number to prevent having too much data stored on the hard disk for an operational company number. Only the product variants that are no longer used in a budget, project, sales order, or quotation can be archived. Before product variant statistics can be printed with the Print Product Variant Statistics (tipcf6410m000) session, the product variants must be archived.
6.3.6 Print Product Variant Statistics (tipcf6410m000)

Product variants that have been delivered to a customer and archived with the Archive Product Variants (tipcf5290m000) session can be printed with this session. The report provides a statistical overview for each generic item, indicating how often a particular option of a product feature has been selected during a specific period.

This information is useful for recording planning percentages for components in the generic BOM. These planning percentages are used to correctly control the planning process in Enterprise Planning.
7. The procedure to define a product model

This chapter describes the main functions of the procedure to define a product model. The objective of this chapter is to explain how and when the procedure is used, the order in which the steps need to be followed, and when the steps are mandatory and/or optional.

7.1 The procedure to define a product model and its result

Companies that assemble to order have problems related to the quantity, complexity, and manageability of the product data and the need for timely availability of the information, if a traditional information system is being used. The answer to this problem is configuration management.

A configuration method can be used to generate all variants of a product. If all the possible components of the bills of materials and all the possible operations for the routing are entered, every product structure of a variant can be created.

Product features are used as the questions that need to be asked of the customer. The answers are used to create the product variant. This is done with decision rules, called constraints. Constraints control the translation of the options selected by the into the product structure of a variant.

Item data can be generated in the same way as the product structure. With constraints, it becomes possible to generate an item code, a description, or even a sales quotation text.

In the price list, the constraints control that a sales price is generated based on the options selected by the customer.

In Heading 7.2, the procedure used to define a product model is described. Heading 7.3 explains the procedure used to test the created product model in different stages of the implementation. It is useful to combine the procedures in these two paragraphs, if the PCF module is implemented.

The result of these procedures is a tested product model that can be used in the operational situation to configure the assemble to order products.
7.2 The procedure to define a product model

Figure 7-1 The procedure to define a product model
7.2.1 Model on paper

Before executing the procedure steps, it is important to make a model on paper. A BOM structure and the routing structure may become too complex if the matter has been given insufficient thought.

The generic BOM and routing form the basic data. The rest of the PCF module is built on this data.

There are a number of issues that need to be considered in this phase:

- The BOM structure and the selection of the generic items, item codes, and item base - and production data.
- It is important is to determine which items need to be created. The main item must be generic. If selections need to be made from alternate items on a lower level, the items on the higher level also must be generic.

To determine the order system of generic items, the following guidelines can be used:

- The generic item with an independent demand can be MPS controlled. Independent means that the demand is not determined by a demand from another item in the same BOM structure.
- The generic items with a dependent demand can be MRP controlled. Dependent means that the demand is determined by the generic main item.

It is also important think about the routing structure. The codes of tasks, work centers, and machines that are used in the routing must be defined in the ROU module.

The following checklist can be used to summarize the parts that need to be included in a model on paper:

- Draw the BOM structure
- Think about the item codes
- Define the generic items
- Determine the order system of the generic items
- Draw the routing structure
- Define work centers, tasks, and machines

After the model is on paper, the system is used to work out the plan. It can be useful to create the product model in a demo company, so that no mix up arises between operational and test data. If the model is created, tested (as described in Heading 7.3), and all the errors are removed, the product model can be copied to an operational company number.
7.2.2 The steps of the procedure

The first step of the procedure is to determine the version of the Product Configurator. If the model is in the creation stage, the interpreter version can be used. The advantage of this version is that the generic product model can be tested immediately if new constraints are created or changes need to be made to these constraints, without first having to recompile all constraints.

The only situation in which the object version must be used in the creation stage of the model is if the extensive features of the Programmers Reference Manual (document number 5555US) are used. This is because the lines in the constraints that use this programming are not interpreted in the interpreter version. If the object version must be used, the constraints must be compiled with the Compile Constraints by Generic Item (tipcf2201m000) session.

If generic items are used on more than one level in the BOM structure, the Multi level PCF Choice Structure parameter in step a must be set to yes to ensure that the question and answer scenario is used on different levels.

The second step in the procedure is entering the generic items (step b) that are needed for the product model. In step c, the required product features are entered. In this session, all product features with possible options are recorded. It is useful to define a self-explanatory code for the product feature, because it improves the readability when they are used in constraints. This is also the case for the product variant statistics, where the codes are used to analyze how often the options of a certain feature are selected.

The product features are associated to a generic item in step d. Product features are controlled by the constraints, which are defined within step f. Messages that contain information for the user can be entered by using these constraints. These messages can be defined in step e, where codes are defined. The codes are used in step f. It is also possible to enter text in the constraint. The text is used as a message without defining it in step e.

Section 3.2.1 describes constraint functionality and shows in which situations that functionality is used.

In steps g and h, the generic product structure and routing can be entered. The constraints (step i) are used to ensure that the product structure and routing are in accordance with the selected options.

If a sales price (or purchase price for generic purchase items) must be generated, the generic price list (step l) can be used to define a price list.

It is possible to use matrices for different features that have mutual relationships associated to the price. By defining the price list code (step j) and the price list matrices (step k), the features and values in the matrices can be entered. Depending how the matrix is used within the generic price list (step l), the values in the matrices represent prices or price factors.
Step m is used if a company wants to use the information given by the customer to create an item code, item description, material, size, text, or standard for a generic item. The generic price lists and item data are controlled by the constraints (step i).

If the product model is defined, the product configurator version is changed from interpreter to object in the PCF Parameters (tipcf0100m000) session (step o). The object version results in a considerable performance improvement compared to the interpreter version. The last step in the procedure is to compile the constraints (step o), so that objects are generated for each item.

If errors are found when the system compiles constraints, they are logged in a system file with a name similar to that of the constraint source file; the name is assigned the suffix .e. This error file is also stored in the same directory as the constraint source file. If there are compilation errors, it may be useful to examine the generated constraint source file. After the errors are removed, the product model can be used in the operational company number.

7.3 System test of the product model

In this heading, the steps used to test the product model in three different stages are described. The first check is to test the product features that are needed to define the product variant for the customer. In the second test, a check is made to determine whether the selected options will result in the correct product structure. The last test determines whether the sales price is in accordance with the selected product options.

7.3.1 System test of the product features

The test for the product features is used to determine whether the product model is in accordance with the operational situation. This test will be executed after the steps a through f are executed in the procedure in heading 7.2.2. This test is important for determining if:

- the question list is complete and in the correct sequence
- no superfluous questions are asked
- all the given options are possible
- the setup of features is user-friendly

The Product Variants (tipcf5501m000) session can be used to test these points.
7.3.2 System test of the product structure

The test for the product structure is used to determine whether the selected options will result in a product variant with the correct product structure, routing, and item data. This test is executed if the generic BOM (step g), the generic routing (step h), and, if used, the generic item data (step m) are defined in the procedure in Heading 7.2.2.

The first step is to determine whether the product structure defines a product variant (step a). This step is optional because this functionality can also be executed in step d with the Product Configurator (tipcf5120m000) session. In step b, a project can be defined that will be used for the product structure of the product variant.

In step c, the project is entered as the project for which a product variant structure must be generated. The product configurator (step d) can be used in the Generate Product Variant Structure for Project (tipcs2220m000) session (step c) to start up the question and answer scenario. This only occurs if step a was not used.

Now, the product variant structure can be generated. By using the print session for customized BOMs (step e) the product structure with routing is available. It is also possible to print the selected product variant options on the same report.
7.3.3 System test of the price list

The test for the price list is used to determine whether the selected options will result in the correct sales price. This test is executed if the generic price lists (step l) are defined in the procedure in Heading 7.2.2.

In step a, the product variant can be entered with the selected options. The sales price for this product variant can be calculated for the product variant in step b. The last step in the procedure is to display the sales price with the Product Variant Sales Price Structure (tipcf5530m000) session(step c). In this report the total sales price is split into the different lines as defined in the generic price list.
The procedure to define a product model
8. The procedure to use a product model in the Sales Order Management module

This chapter describes the main functions of the procedure to use a product model in the Sales Order Management module. The objective of this chapter is to explain how and when the procedure is used, the order in which the steps need to be followed, and when the steps are mandatory and/or optional.

8.1 The procedure to use a product model in the SLS module and its result

Under the condition that no errors are present in the product model, the advantages of using a product model within sales are:

- The sales person does not need any knowledge about the product and the associated sales prices. He/she only needs to ask the questions and provide possible options to the customer. The product variant is defined by entering the selected options.

- It is not possible to offer a product to the customer that cannot be manufactured, from a technical production point of view.

- A direct translation occurs from the commercial offer to a product structure that is used in the assemble to order situation.

In a quotation stage, a product model can be used to define the product the customer wants. A sales price can be generated based on the selected options. The question and answer scenario makes it simple to translate the requirements of the customer into a product structure of the variant by selecting the options the customer wants. By creating a budget for the request, the cost price of the product can be calculated.

The result of using PCF for a sales quotation is that a product variant is created in the quotation, a product structure is generated for a budget, and the budget is used to calculate the cost price of the product.

If the product model is used in a sales order, a product variant with sales price is generated based on the options selected by the customer. By creating a project, the product can be manufactured.

The result of using PCF in a sales order is that a product variant is created in the order, a product structure is generated for a project, and the project is used to manufacture the product variant.
8.2 The procedure to use a product model in a sales quotation

If the customer wants a quotation for a product that must be assembled to order, a product model can be used to define the product variant. By executing step a, the sales quotation is entered.

In the order lines of the sales quotation, the product configurator (step b) is used to start the question and answer scenario by selecting the generic item. The options selected by the customer are entered in the product configurator. In the order lines, the sales price is generated. If information about the sales price must be provided to the customer, it can be displayed in a report that shows the different price list lines (step c).

The display of the product variant sales price structure can be used as a zoom session from the Product Variant field in the sales quotation line.

Figure 8-1 The procedure to use a product model in a sales quotation

In the order lines of the sales quotation, the product configurator (step b) is used to start the question and answer scenario by selecting the generic item. The options selected by the customer are entered in the product configurator. In the order lines, the sales price is generated. If information about the sales price must be provided to the customer, it can be displayed in a report that shows the different price list lines (step c).

The display of the product variant sales price structure can be used as a zoom session from the Product Variant field in the sales quotation line.
The quotation is printed in step e. This step can be executed before or after the product variant structure is generated for a sales quotation (step d). If the quotation is printed after step d, then the item code, description and text are generated by the generic item data within the PCF module and printed on the quotation. A condition to printing the sales quotation text is that the Generate Sales Quotation Texts parameter in the PCF Parameters (tipcf0100s000) session is set to yes.

If the quotation is printed before the product variant structure is generated, the generic item code, description, and text are not included on the quotation.

If a cost price must be calculated to determine if the desired sales margin is achieved, a product variant structure can be generated for a sales quotation. This structure is generated to a budget. In this budget, it is possible to obtain insight into the cost price of the product variant offered to the customer.

8.3 The procedure to use a product model in a sales order

![Diagram](image)

**Figure 8-2 The procedure to use a product model in a sales order**

If the customer orders a product that must be assembled to order, a product model can be used to define the product variant. By executing step a, the sales order is entered.
By selecting a generic item in the order lines of the sales order, the product configurator (step b) can be used for the product model. The options selected by the customer are entered within the product configurator. The sales price is generated in the order lines.

If information must be provided to the customer about the sales price, it can be displayed in a report that shows the different price list lines (step c).

The display of the product variant sales price structure can be used as a zoom session from the Product Variant field in the sales order line.

An order acknowledgment is printed in step e. This step can be executed before or after the product variant structure is generated for a sales order (step d). If the order acknowledgment is printed after step d, then the item code, description, and text are generated by the generic item data in the PCF module and printed on the acknowledgment.

If the acknowledgment is printed before the product variant structure is generated, the generic item code, description, and text are not included on the order acknowledgment.

By executing step d the product variant structure is generated from the product variant of the sales order into the product structure of a customized item that is always a part of a project. The project makes it possible to produce this product variant.

A description of the use of projects can be found in the material for PCS Projects and for the ERP 5.0b Project package.
Appendix A

Examples of constraints

Because constraints control the entire product model, five examples are included in this appendix. These examples provide additional information about the use of constraints. In addition, programming examples are also provided.

This appendix is not intended as a comprehensive treatment on constraints. For additional information, refer to the on line help text in the Product Constraints business object and the Programmers Reference Manual.

The examples are based on a bicycle manufacturer. Different types of bikes are sold (this is [typebike] within the constraints). One of these types is a mountain bike (mtanbike). During the question and answer scenario, a question is asked about the number of gears (num_gear) the customer wants on the bike.

In every example, the first two lines indicate where the constraint is used (by the constraint ID: f for features, o for routing, i for item data, and p for price list) and which constraint section is used.

The constraint is provided after these lines. The numbers at the beginning of each line are only used to make it easier to refer to a particular line.

After the constraint lines an explanation is provided about the purpose of the constraint. The contents of the constraint lines are also explained.

Example I

- Constraint ID: f001  How many gears do you want?
- Constraint Section: Before input

1 input = false
2 display = false
3 [num_gear] = 3
4 if [typebike] = “mtanbike” then
5   input = true
6   display = true
7 endif

Explanation of example I

This constraint is used for a feature. If the customer wants a mountain bike, the customer can tell how many gears(s) he wants on the bicycle. In every other situation, three gears are installed and the question is not displayed in the Product Configurator.

1 No value can be entered in the product configurator
2 The question is not displayed in the product configurator
3 Three gears are the default value
4 If the customer wants a mountain bike
5 a value can be entered in the product configurator
6 and the question is displayed
7 End of the statement
**Example II**

- Constraint ID: f001  How many gears do you want?
- Constraint Section: Validation

```plaintext
1 if [num_gear] > 0 and [num_gear] < 22 then
2     validate = true
3   else
4     validate = false
5   message = “ We do not deliver bikes with more than 21 gears ! “
6 endif
```

**Explanation of example II**
The maximum number of gears a customer can order is 21. The constraint is used for the same feature as in example I.

1 The number of gears must be more than zero (at least one gear is delivered) and can not be more than 21.
2 If that is the case, the answer is allowed
3 else
4 the answer is not allowed
5 and the user of the system will get a message
6 End of statement

The same constraint can be defined by entering the following sentence.

c: [num_gear] > 0 and [num_gear] < 22

This line performs exactly the same as the first six lines. The only difference is that the message is not given to the user of the system because a message can not be used if the if-then-else statement is written as c:.

**Example III**

- Constraint ID: i001  Description of the desired bike
- Constraint Section: Parameter Substitution

```plaintext
1 item = “type “ & [typebike] & ” ” & str( [num_gear]) & ” ” gear(s)”
```

**Explanation of example III**
The description of the generic item is generated based on the answers given by the customer to the questions: type of bike and number of gears. The constraint is used in the Generic Item Data business object.

1 The description uses two words, independent of the selected options. These words are type and gear(s). The answers given to the type of bike and number of gears are used to make a description. Because the option domain of the feature [num_gear] is not alphanumeric, a string must be used. If the following answers are given: [typebike] = mtanbike, [num_gear] = 12, the description of the item is: type mtanbike 12 gears.
**Example IV**

- Constraint ID: o001  The task: Assembly
- Constraint Section: Parameter Substitution

```plaintext
1 run_time = 60
2 setup_time = 15
3 if [typebike] = “mtanbike” then
4   run_time = 85
5   setup_time = 15
6 endif
```

**Explanation of example IV**

If the customer wants a bicycle the run time to assemble the bike is 60 minutes and the machine set-up time of the is 15 minutes. If the customer wants a mountain bike, the operation time to assemble this type of bike takes 25 minutes longer. No change is made to the set-up time of the machine. The constraint is used in the Generic Routing business object.

```plaintext
1 The default operation time to assemble a bicycle is 60 minutes
2 The default set-up time for the machine is 15 minutes
3 If the customer wants a mountain bike
4 the operation to assemble this bike is 85 minutes
5 the set-up time for the machine is still 15 minutes
6 End of statement
```

**Example V**

- Constraint ID: p001  Surcharge based on number of gears
- Constraint Section: Parameter Substitution

```plaintext
1 if [num_gear] is { 9,11, 13 -21} then
2   validate = true
3 else
4   validate = false
5 endif
```

**Explanation of example V**

In some cases, the number of gears ordered will change the operation time used to manufacture the bicycle. By defining a surcharge, the extra costs are covered. The constraint is used in the Generic Price List business object.

```plaintext
1 If the customer wants a bike with 9 or 11 gears, a surcharge is added to the sales price. This is also the case if the customer orders 13, 14, 15, 16,17, 18, 19, 20 or 21 gears.
2 If the customer orders a number of gears as stated in the first line, the surcharge is added to the sales price
3 else
4 no surcharge is added
5 End of statement
```
Examples of constraints