

EXPERTIZERS : ANALYSIS OF MODEL 'EXAMPLE'



Model example.ex1
Step 3 of your learning
Description of the branch 'operations' and how it works
Written by Didier Riche, designer of the software

Before proceeding to this step, it is imperative to have followed the first learning steps 1 and 2 of the case EUROTOP

Expertizers Copyright 2006-2009

Where are you after steps 1 and 2 ?

The two steps of the case EUROTOP have enabled you to familiarize yourself with good numbers of concepts and manipulations.

You know how to manage the relationship between the structure of a file of events, and its description in a model.

You know how to use the distribution functions, create variables with checklists values or rules, create variables decode and sum_. You know the update, delete, and even rename functions.

You know how to create activities, and formulate rules of condition (where) and calculation (*variable =*)

In short, you already know a lot !

It only remains for you to discover the huge potential offered by the additional branch 'OPERATIONS' ... and some other functions.

The branch 'OPERATIONS' is not restricted to industrial problems. Many administrative tasks using time-driven method (in a call center for example) have recently been treated in this way. The software accepts that operations and activities use the same code.



Presentation of events to be processed: Orders of products characterized by various criteria, such as their diameter

Unlocked mode		Event file description									
Variable Code	Fixed Var.	Variable Name	Format	Tot. Length	Decim	Batch	Quot	R. T	Col	Values to Control	
entity	X		varchar	20	0F	F		0		PARIS/	
period	X	accepted formats are YYYY or YYYY/MM or YYYY/MM/DD	varchar	10	0F	0					
product_code	X		varchar	20	0F	N		0			
event_code	X		varchar	20	0F	F		0			
diameter			integer	3	0F	0		0		10/20/30/	
machine			varchar	7	0F	F		0		LINE_1/LINE_2/	
quantity_init			numeric	16	30	0		0			
customer_name			varchar	30	0F	F		0			
country			varchar	3	00	0		0		France ~tFR/England ~tGB/	

Here in quotation mode, some data are compulsory (0 parameter). Thus their typing is under control.

You find the data in the path and file: c:\Expertizers \Import \EVENT_TEST_1.txt

PARIS	2006/01/15	GROUP_1		20		10000	SMITH	GB
PARIS	2006/01/15	GROUP_2		20		50000	DUPONT	FR
PARIS	2006/01/15	GROUP_2		10		5000	DUPOND	FR
PARIS	2006/01/15	GROUP_1		10	LINE_1	100000	LEE	US
PARIS	2006/01/16	GROUP_1		20		15000	ZELLER	GY
PARIS	2006/01/16	GROUP_1		20		5000	MORGAN	US
PARIS	2006/01/17	GROUP_1		20		50000	MARTINI	US

Visual presentation of the operations branch of the model

The screenshot shows the 'Batch side - Settings' window. On the left is a tree view under 'Root' with the following structure:

- Variable Types
 - Event File Entry Description (Updatable)
 - Cost Activity File Entry Description (Not Updatable)
 - User Internal Variables (Updatable)
 - Variables generated automatically
- Entity Description
 - PARIS
 - Operations
 - PACKAGING
 - PACK_10
 - 10 cadence = 50
 - 20 quantity = quantity * 1.05
 - PACK_20
 - 10 cadence = 35
 - 20 quantity = quantity * 1.07
 - PRODUCTION
 - LINE_1
 - 10 cadence = 100
 - 20 cadence = 70
 - 30 quantity = quantity * 1.1
 - 40 quantity = quantity * 1.2
 - LINE_2
 - 10 cadence = 80
 - 20 cadence = 75
 - 30 quantity = if diameter = 10 then

On the right is a table titled 'operations' with the following data:

Operation	D/X	Prod	Condition 1	OR Condition 2
PACKAGING		Y		
PRODUCTION		Y		

Under the entity PARIS, the model describes two operations (which can be just as workshops or administrative services). The first is 'PACKAGING'. The two types of machine PACK_10 and PACK_20 are characterized by rules of tempo and waste related to the complexity of the products to be manufactured. PRODUCTION The second operation, shows the same structure: Objects which are machines LINE_1 and LINE_2 and below each one, none limited object lines.

Important Note: operations must always be presented in reverse order of the normal cycle of processing. So, the packaging is described before the manufacturing process. You will know why when studying the use of _init variables.

All operations have not need to be described in the branch operations. If the processes associated with an activity are simple, just describe the rule at the activity level. By opening the branch activities, you will see that is the case for some of the model.

Introduction to variables termination _init

The termination _init variables are in two places: in the section 'File Event Entry Description ' and in the section 'Internal User variables'.

Two variables quantity_init and cadence_init are always available in the basic model.

You cannot delete them, for against you can rename, move and add more if needed (and deletable as at least one remains in the section).

In both steps of EUROTOP case you used the variable quantity_init (you had requirement; already available why not use it), but without the characteristics they take in their association with the OPERATIONS.

You will see that according to section membership, variables termination _INIT react not in the same way, but both in their formulation generic _INIT come in several variables generated automatically in The section 'variables generated automatically'.

Quantity_init : quantity, quantity_ope_OPERATION (*as many as operations created*)

Cadence_init : cadence, cadence_ope_OPERATION



Viewing an operation and its objects

To create the first operation, click the Operations section, and then right click mouse option 'New sub-level'.
 To create another operation, click on the operation on which you want to insert below, then right click mouse, 'insert' option.

Unlocked mode

Operation and objects below

Operation Code and Name: PRODUCTION production lines

Production: Yes Comments.....

If no condition is provided, all event lines of the entity PARIS (or with entity missing) will enter the operation

{WHERE} condition 1. Doubleclick for help (OR) condition 2. Doubleclick for help

D/X: No Limit From To

facultative reference to other operation to benefit with its objects and rules. Itself by automatic default. PRODUCTION

Object	D/X	How M	Condition 1	OR Condition 2
LINE_1		2,00	machine = ''	machine = 'LINE_1'
LINE_2		3,00	machine = ''	machine = 'LINE_2'

The level 'object of operation' being mandatory, the option 'No' automatically generates an object with the same code as the operation code

If the event line is not dealt with reference to a machine, the event will enter the two objects. For if the line against specifically mentions a machine (eg LINE_2) the event will only enter in object LINE_2



Viewing an object and its property lines (click on object)

Viewing a property line (click on line)

Unlocked mode

Object and lines below

Object Code + Name:

How Many Similar: **number of identical machines**

(Where) condition 1: Doubleclick for help
 machine = "

(OR) condition 2: Doubleclick for help
 machine = 'LINE_1'

D/X: From To

To create the first object under an operation, click on the operation then right click mouse option 'New sub-level'. To create another object, click on the object on which you want to insert below, then right click mouse, 'insert' option.

Line	D/X	Condition	Rule
10			cadence = 100
20		diameter = 20	cadence = 70
30		diameter = 10	quantity = quantity * 1.1
40		diameter = 20	quantity = quantity * 1.2

Entity Description

PARIS

Operations

PACKAGING

PRODUCTION

LINE_1

10 cadence = 100

20 cadence = 70

30 quantity = quantity *

Where Condition diameter = 10

Variable = Rule quantity = quantity * 1.1

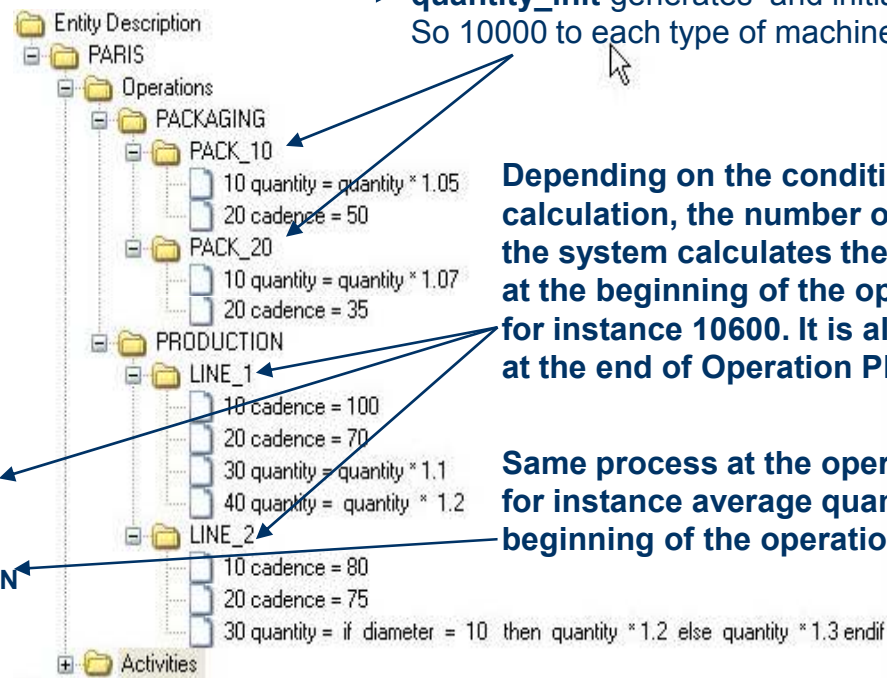
To create the first line under an object, click on the object then right click mouse option 'New sub-level'. To create another line, click on the line on which you want to insert below, then right click mouse, 'insert' option.



Characteristics of variables termination `_init` defined inside the 'event file description' section

Focus on the order line 1:

Diameter	Machine	Quantity Init
20		10000,000
20		50000,000
10		5000,000
10LINE_1		100000,000



`quantity_init` generates and initializes the `quantity` variable; So 10000 to each type of machine

Depending on the condition of rules, rules calculation, the number of machines of the same type, the system calculates the **average quantity** needed at the beginning of the operation **PACKAGING**: for instance 10600. It is also the quantity to be produced at the end of Operation **PRODUCTION**

Same process at the operation **PRODUCTION**: for instance average quantity = 11,500 needed at the beginning of the operation **PRODUCTION**

`Quantity_ope_PACKAGING`
10600

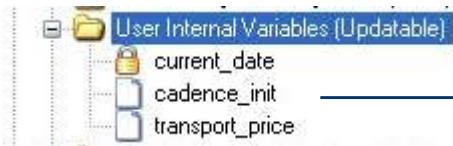
`Quantity_ope_PRODUCTION`
11500

Then, these variables are ready to be used in **ACTIVITY** rules as well as `quantity_init` which keeps the initial value. But never use the variable `quantity` in **ACTIVITY** rules.

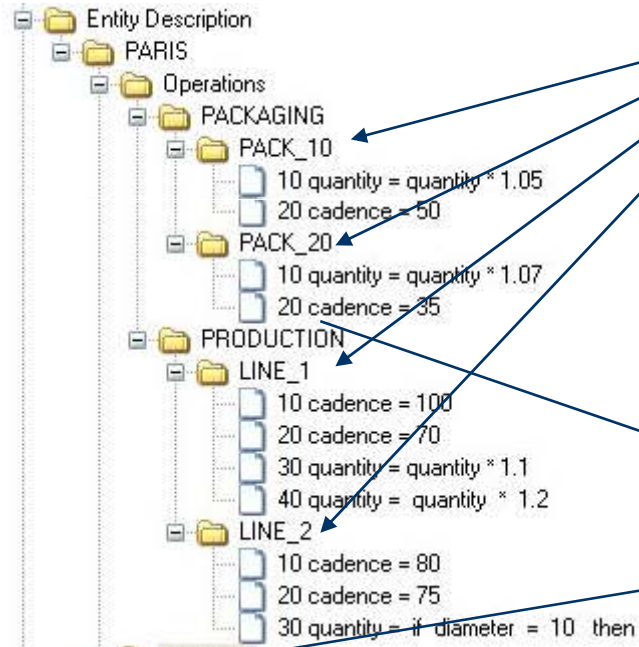
This method of calculation explains why your operations must always be presented in reverse order of the normal cycle of processing, so from the sale to purchase.



Characteristics of variables termination `_init` defined inside the 'User Internal Variables' section



`cadence_init` generates the variable `cadence`. Before treating each object (here type of machine), the variable `cadence_init` is reset to zero. (never use `cadence_init` in the rules, but just `cadence` at this level)



Before leaving each operation, according to the rules of condition, the calculation rules, the number of objects of the same type, the system calculates the average `cadence` of each operation for the event line.

`cadence_ope_PACKAGING`

`cadence_ope_PRODUCTION`

Then, these variables are ready to be used in **ACTIVITY** rules.

But never use the variables `cadence_init` and `cadence` in **ACTIVITY** rules.



Display of activities and their rules

Activity Code	Name	B/D	Mem	Re	Where ...	Activity_Value=	Cost_Value=	Text_Value=
MATERIAL	raw material cost	N	Y			activity_value = quantity_ope_PRODUCTION	activity_cost = activity_value * 0.12	activity_text=PL
PRODUCTION	Production lines	N	Y			activity_value = if cadence_ope_PRODUCTION = 0 then 0 else (quantity_ope_PRODUCTION / cadence_ope_PRODUCTION) endif	activity_cost = activity_value * if period < '2006/01/17' then 10.45 else 11 endif	
PACKAGING	packaging and control step	N	Y			activity_value = if cadence_ope_PACKAGING = 0 then 0 else quantity_ope_PACKAGING / cadence_ope_PACKAGING endif	activity_cost= activity_value * 9.75	
RESALE_RAW_MAT	recuperation of waste products	N	Y			activity_value = quantity_ope_PRODUCTION - quantity_init	activity_cost = activity_value * -0.10	
TRANSPORT	delivery to the customer	N	Y			activity_value = quantity_init	activity_cost = (activity_value / 10000) * transport_price	
ERROR_DATE	check if period is correct	N	N			activity_value = if ISDATE(period) = 1 then 0 else 1 endif		
TVA		N	Y			activity_value = cum_cost	activity_cost = (activity_value * 19.6) / 100	

- able Types
- Event File Entry Description (Updatable)
- Cost Activity File Entry Description (Not Updatable)
- User Internal Variables (Updatable)
- Variables generated automatically
- Activity Description
- PARIS
 - Operations
 - Activities
 - MATERIAL
 - PRODUCTION
 - PACKAGING
 - RESALE_RAW_MAT
 - TRANSPORT
 - ERROR_DATE
 - TVA
- Station Data Order

Activity ERROR_DATE
Explained pages following

Variable cum_cost
Explained pages following

cost_value column.
values the work units
by standard cost or
constants for quotation



Characteristics of activities: ERROR_

Any activity starting by ERROR_ and where the value of the variable activity_value is different from zero, will generate a list of errors. It is a mean to provide consistency checks.

This can go very far to write sophisticated rules. For this reason, the software is also suitable for analysis of surveys, sampling, and control the content of any file.

The detection of errors leads to the automatic display of them.

Options for conservation or lines of erroneous events are available on the path: Menu / Batch side / Error parameters.

The first option is related to errors intrinsic values (mandatory, checklist values, ...) found during batch loading. The second option is related to errors generated by activities like ERROR_ found during calculation process.

- Keep events with errors during batch loading
- Keep events with errors during batch calculation

Characteristics of CUM_COST variable

CUM_COST variable is the total of all values activity_cost , reached at the point it is used in an activity rule .

Activity lines excluded from the result (option 'Display in results' set to No) are therefore excluded.

Complements

The model Example.ex1 can operate with two different presentation formats of events.

The model presented is an event-by-line (file EVENT_TEST_1.txt).

You can also use the file EVENT_TEST_2.txt (an event with more than one line, each line with a different record type). But before you load it correctly you will need to import data (Menu / import) from TEST_2.TXT which adds the variable 'record_type' at the fifth position in the 'Event File Entry Description' section. This example serves to understand how to describe your event if you are faced with this need.

You can load external costs (file COST_TEST_1.txt). Note that if the activities do not marry with all the carriers of external costs, standard costs calculated of non-married are preserved.

At this stage of learning you are now able to achieve many applications. If you have other needs, you will find in the documents of training also available on the site www.expertizers.com

Other Items: catalog management; group management of materials and accessories;
Administration / distribution of models



Warning

This document is brought to you by DB.SOFTART.SERVICES. (« DB »).

This document and its contents are the product of research and important development led by DB.

They are presented solely in order to allow you to evaluate some of EXPERTIZERS software's capabilities

You must process the content of this presentation and any other information or document, in connection with the solutions presented during any lecture, as confidential and exclusive property of DB.

and receiving the information, you agree to use the information for the sole purpose of evaluating your own account

for the software's capabilities Expertizers to maintain confidentiality and not to transmit to anyone, all or part of the information without prior written permission of DB.

- DB.SOFTART.SERVICES
Didier RICHE
Software Publisher
+33 (0)6.63.92.20.60
+33 (0)1.47.34.40.23