

StanForD  
Created 2001-12-18  
Last rev 2006-05-29

# **Extended documentation of StanForD-variables**

Preliminary document

# Table of contents

1	Introduction .....	1
2	Variable 5, VARIANT.....	1
3	Variable 22 - FILEORDER.....	1
4	Variable 60 - WEIGHTCAL .....	4
5	Variable 105 – GRADESYS.....	5
6	Variable 113 – BARKPAR.....	6
7	Variable 132 - LWRLNGTHLMT' .....	6
8	Variable 190 – MAXLOG .....	8
8.1	Type 1 .....	8
8.2	Type 2 .....	9
8.3	Example.....	9
8.3.1	Code = 1 .....	9
8.3.2	Code = 3 .....	9
8.3.3	Code = 5 .....	10
8.3.4	Code = 7 .....	10
9	Variable 196 – OVERPROD .....	11
9.1	Type 1 .....	11
9.2	Example.....	11
10	Variable 197 – FROMMATRIX .....	12
11	Variables 221 - NUMSTMS and 290 - NUMLOGS.....	13
12	Variable 241 - TOTMERCVOL and Variable 249 – TOTALVOL.....	13
12.1	Historical background.....	14
12.2	Description.....	14
13	Variable 244 – OUTSPECVOL.....	14
14	Variable 503 - NUMSTEMDB .....	16
15	Variable 523 - COORDSTM.....	17
16	Revisions of document .....	19
17	Index.....	20

# 1 Introduction

This document is a complement to the official StanForD documents. It contains more detailed descriptions of some of the variables as well as examples of how the variables can be used. The document will also contain other information related to StanForD, for example general description of different file types.

The document is to a large extent based on the background material used for making the decisions at the StandForD meetings. The secretary can update and publish this document at any time between meetings.

## 2 Variable 5, VARIANT

In October 2002 Ola Söderlund, Partek, raised the issue of how to handle different versions of StanForD. The proposal was partly based on his ideas.

It is today not possible to determine which version (date) of StanForD has been implemented in a specific file from a specific software. If this was possible it would simplify the possibility to check whether a file follows the standard or not as well as quickly determine whether a file includes certain variables or not. It will improve the possibility for users to keep track of how and when different versions of StanForD are implemented.

It should be possible to keep track of StanForD-versions implemented by recording the version in all StanForD-files.

The following new variable was therefore introduced (2003-04-08) with priority 2 in all files:

Var	Name	Type	Data type	Unit	Description
5	VARIANT	3	string	yyyymmdd	Version of StanForD when last implementation of new or updated StandForD variables in present file was carried out. Version of StanForD is date when last update of standard was decided e.g. 20030408.

## 3 Variable 22 - FILEORDER

FILEORDER was introduced at the meeting in October 2002 (2002-10-28). The descriptions of variable 19 (BREAKDATE) and 20 (RESTARTDATE) were updated at the meeting in April 2002. However, these new descriptions make it impossible to use these variables if we, after a restart, want to store data in a totally new production file (prd- or pri-file).

In some cases it is needed to restart on a totally new production file as described by the following two examples:

1. If the operator has to change an apt-file during harvesting of a certain site he might be forced to restart on a new prd-file. For example because the limits of the length classes of a certain assortment have been changed.
2. If the operator has to send a pri-file everyday and he wants to avoid having to send already reported log data every time he is sending the pri-file. This will mean sending significantly smaller files.

The following new variable was therefore introduced (priority 3 in prd-files and priority 1 pri-files):

Var	Name	Type	Data type	Unit	Description
22	FILEORDER	1	integer	integer	Order of production files (prd or pri), used when dividing production result from one site into several different files. In the first file FILEORDER = 1. Only the last file has an ENDDATE (var17_t4). When production is resumed and production is stored in a new file, with no old data, FILEORDER must have a value >1 <b>The file is considered to be cumulative if variable is missing</b>

If the total production is registered in one single file variable FILEORDER is equal to 1 and the file has an ENDDATE. RESETDATE and STARTDATE will also be identical.

The last and final production file from a site must have both a FILEORDER number as well as an ENDDATE. If a file has a FILEORDER-value large than 1 and no ENDDATE (var17\_t4) then several files have to be summed up in order to calculate the total result of a site. All FILEORDER values between 1 and n must be included when summing several files, where n=the FILEORDER number of the last file, the last file has an ENDDATE value.

To keep track of the different pri-files the variable FILEORDER (var22\_t1) should be used. The variable is used when dividing production result from one site into several different files (non-cumulative). In the first file FILEORDER is equal to 1. Only the last file has an ENDDATE (var17\_t4). When production is resumed and production is stored in a new file, with no old data, FILEORDER must have a value larger than 1. If the total production is registered in one single file variable FILEORDER is equal to 1 and the file has an ENDDATE.

The last and final production file from a site must have both a FILEORDER number as well as an ENDDATE. If a file has a FILEORDER-value large than 1 and no ENDDATE (var17\_t4) then several files have to be summed up in order to calculate the total result of a site. All FILEORDER values between 1 and n must be included when summing several files, where n is equal to the FILEORDER number of the last file. The implementation of FILEORDER when storing data in a non-cumulative file is illustrated in figure 2.

*Not cumulative = file is reset after each reporting, meaning that data is reported only once.*

Prod data for 0-25%		Prod data for 25-60%		Prod data for 60-100%	
FILEORDER	1	FILEORDER	2	FILEORDER	3
STARTDATE	01, 7.00	STARTDATE	01, 7.00	STARTDATE	01, 7.00
RESEDATE	01, 7.00	RESEDATE	02, 7.00	RESEDATE	03, 7.00
SAVEDATE	01, 18.00	SAVEDATE	02, 18.00	SAVEDATE	03, 18.00
ENDDATE	-	ENDDATE	-	ENDDATE	03, 18.00
VarSum	25%	VarSum	60%	VarSum	100%

Figure 2) Example illustrating non-cumulative files, the total production is calculated by summing file 1, 2 and 3.

However, it may in some cases be necessary to use the pri-file as a cumulative file. A cumulative file is defined as a file which is not reset between reporting, meaning that each day's production is added to the same pri-file. In that case FILEORDER (var22\_t1) should have number 1, while ENDDATE (var17) is non-existing or empty until the production at the site is totally finished. If a pri-file has a FILEORDER value of 1 as well as an ENDDATE it will contain the total production results from a certain site. No summing of pri-files will be needed in this case. Observe that if FILEORDER is missing the file is considered to be cumulative. The implementation of FILEORDER when storing data in a cumulative file is illustrated in figure 2.

*Cumulative files = file is not reset after reporting, all data from a site at the time of reporting is stored in the file*

Prod data for 0-25%		Prod data for 0-60%		Prod data for 0-100%	
FILEORDER	1	FILEORDER	1	FILEORDER	1
STARTDATE	01, 7.00	STARTDATE	01, 7.00	STARTDATE	01, 7.00
RESEDATE	01, 7.00	RESEDATE	01, 7.00	RESEDATE	01, 7.00
SAVEDATE	01, 18.00	SAVEDATE	02, 18.00	SAVEDATE	03, 18.00
ENDDATE	-	ENDDATE	-	ENDDATE	03, 18.00
VarSum	25%	VarSum	60%	VarSum	100%

Figure 3. Example illustrating non-cumulative files, the total production is calculated by summing file 1, 2 and 3.

By mistake (if for example the pri-file is not reset properly after reporting) we may also end up with a case where files are both pri-files are both a cumulative file and non-cumulative. This is illustrated by the following example:

Day	Daily production (no of logs)	No of logs in file	Production stored in pri-file from day	Cumulative file	FILEORDER (var22_t1)	ENDDATE (var17_t4)
1	3100	3100	1	No	1	-
2	2900	2900	2	No	2	-
3	3200	6100	2 and 3	Yes	2	-
4	2800	8900	2, 3 and 4	Yes	2	-
5	2600	2600	4	No	3	01-01-05

To calculate the total production, files from day 1, 5 and 4 (the last file with FILEORDER 2) are all summed, in this example 3100 + 8900 + 2600. This means that the last file with a specific FILEORDER (in this case no FILEORDER 2 from day 4) is always used when calculating the total sum.

In order to check that the correct files have been used to calculate the total results the variables NUMSTEMS (var221\_t2) and NUMLOGS (var290\_t2)

can be used as checksums. These variable are never to be reset while production at a site is not totally finished and they were added in April 2003 (2003-04-08). It is also possible to compare the RESETDATE (var11) and the SAVEDATE (var12). If the previous file reported from the harvester has the same RESETDATE as the present file as well as the same FILEORDER we know that it is a cumulated file and vice versa.

Below is an additional example of how to use the new variables for daily reporting of the production results for three days using a pri-file. Observe that the length of the date and time strings are a bit shortened for practical reasons.

<b>Date and time</b>	<b>Logging operation</b>	<b>Prod. reporting</b>	<b>Action concerning pri-file</b>	<b>Var22 t1 FILE-ORDER</b>	<b>Other variables affected</b>
<b>Day 1</b> 02-10-10, 10.00	Start on new site		Start on new file	1	STARTDATE: 02-10-10, 10.00
<b>Day 1</b> 02-10-10, 22.00	Close for the night	Sending daily pri-file	Sending report (pri-file)		
<b>Day 2</b> 02-10-11, 08.00	Start for the day		Start on new file, old data excluded	2	
<b>Day 2</b> 02-10-11, 19.00	Close for the night	Sending daily pri-file	Sending report (pri-file)		
<b>Day 3</b> 02-10-12, 07.00	Start for the day		Start on new file, old data excluded	3	
<b>Day 3</b> 02-10-12, 15.00	Site finished	Sending final report	Sending report (pri-file)		ENDDATE: 02-10-12, 15.00

*This means that the first pri-file sent on the first day has the following variables:*

STARTDATE (var16 t4)	02-10-10, 10.00
ENDDATE (var17 t4)	Not existing
FILEORDER (var22 t1)	1

*This means that the last pri-file sent on the third day has the following variables:*

STARTDATE (var16 t4)	02-10-10, 10.00
ENDDATE (var17 t4)	02-10-12, 15.00
FILEORDER (var22 t1)	3

To calculate the total result, three files with the following FILEORDER numbers have to be summed: 1+ 2 + 3

## 4 Variable 60 - WEIGHTCAL

Since there are existing systems for weighing the load of a forwarder as well as to record the mass (MASS, var237) it is necessary to be able to record how and when control and calibration of the scale is done.

It was decided at the meeting in October 2002 (proposed by Timberjack) to include a new variable for registering data in the prd-file concerning control and calibration of scales.

The following new variables were decided (2002-10-28) with priority 3 in prd-files:

Var no	Name	Type	Data type	Unit	Description
60	WEIGHTCAL	1	Integer	Integer	Number of scale calibrations
	WEIGHTCAL	2	String	yyyymmdd hhmmss	Date for calibrating scale: 1...var60_t1
	WEIGHTCAL	3	Integer	Kg	Reference mass when calibrating scale: 1...var60_t1
	WEIGHTCAL	4	Integer	1%	Error of scale when calibrating: 1...var60_t1
	WEIGHTCAL	5	Integer	0,001	Correction factor used when weighing, default value is 1000.

Calibration (WEIGHTCAL) is a calibration of a scale. A 1000 kg load (*Reference mass*, var60\_t3) is lifted with the crane and this is compared with the measurement of the scale (mass according to scale) and the scale is corrected with a factor (*Factor used when weighing*, var60\_t5), if correction is needed. The *true mass* (MASS,var237) is registered in the prd-file.

*Factor used in weighing* is used when calculating the *True mass* from the *Mass according to the scale*. If needed the *Factor used in weighing* is adjusted when the scale is calibrated.

$$\text{Factor used when weighing} = 1000 * (\text{Reference mass}) / (\text{Mass according to scale})$$

$$\text{True mass} = (\text{Mass according to scale}) * (\text{Factor used when weighing} * 0.001)$$

$$\text{Error of scale when calibrating} = ((\text{Weighed mass before calibration} - \text{Reference mass when calibrating scale}) / \text{Reference mass when calibrating scale}) * 100$$

Below is an example of a calibration:

$$\text{Reference mass} = 1000 \text{ kg}$$

$$\text{Mass according to scale} = 950 \text{ kg}$$

$$\text{Error of scale} = ((950-1000) / 1000) / 0.01 = 5$$

$$\text{Factor used when weighing} = (1000 / 950) / 0.001 = 1053$$

$$\text{The following calibration variables are registered: var60_t3} = 1000, \text{var60_t4} = 50, \text{var60_t5} = 1053$$

## 5 Variable 105 – GRADESYS

Variable 105 was implemented 1995-06-12. If a price matrix was defined as quality 3 in the old system (code 1 Random grades / Fallande kvaliteter) it meant that any log with quality 3, 2 or 1 could be cut. This was not a good solution when the grading system in Sweden was changed (mid 1990s) so that for example a quality 3 meant that you could only include a log with quality 3 or 1 but NOT quality 2. Thus code 2 (permissible grade/preciserade kvaliteter) was implemented. Code 2 means that each individual quality must be specified

for a certain price matrix. In the case of a price matrix for quality 3 you specify that both quality 1 and quality 3 are allowed. The grades are register in var141.

## 6 Variable 113 – BARKPAR

Different bark reduction parameters are used in Germany (var112\_t2, var113\_t2 and var113\_t3) and Sweden (var112\_t1 and var113\_t1). Four hard coded deduction values 10, 20, 30 and 40 mm are used in Germany. A diameter interval is attached to each deduction value. Depending on the diameter at each position along the stem, one of the four deduction values is valid. The deduction is equal to the double bark thickness, for example deduction 10 mm is equal to 5 mm bark at each side of the log. The diameter break (var113\_t2) is the bottom of the class.

In Germany the deduction is fixed but the interval can be changed.

Example:

Deduction (mm) "Hardcoded" (var113_t2)	Diameter break (mm) Example. (var113_t3)
40	>=600
30	<600 >=400
20	<400 >=200
10	<200 >=0

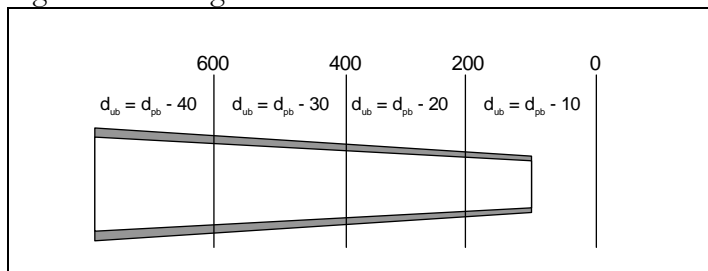
The example above would result in the following variables:

112 2 4 4 4~

113 2 600 400 200 0 600 400 200 0 600 400 200 0~

113 3 4000 3000 2000 1000 4000 3000 2000 1000 4000 3000 2000 1000~

Figure illustrating how diameter under bark is calculated:



The following updates of tThe descriptions of var113\_t2 and var113\_t2 were updated on 2003-10-28.

## 7 Variable 132 - LWRLNGTHLMT

Timo Hokka, Metsäteho, presented the following problem concerning lower length class limit at the meeting in April 2002.



There are two different ways to handle lower length class limits (var 132 t1) combined with cutting window limits. A short example: Length classes 400 cm and 430 cm (var 132 t1) cutting window [-3, 3] (var135 t3, t4), log length 427 cm

- 1) In the first case the cutting window is taken into account and the log is registered in class 430 (Timberjack, Ponsse, Mitron, Mevire).
- 2) In the second case the cutting window is not taken into account, the log is registered in class 400, because it's length is shorter than the lower limit of the lengthclass. (Dasa, Valmet) In practice negative cutting window is not used with these machines but it's possible to use it.

It should be noted that no overlapping between the cutting windows of different length classes is allowed, that is the cutting window cannot be wider than a length class. The following new descriptions of variable 135, OVERSIZE, were accepted at the meeting 2002-10-28, the definitions are also illustrated in figure 1-3:

Var no	Type	Description
135	1	Additional length margin/length class/price matrix/tree species, <b>can not be a negative number:</b> 1..var116/1..var111
135	3	Lower <b>length</b> limit for "cutting window"/price matrix/tree species. <b>Lower length class limit (var132) and variable 135, type 1 and 3 together, define the length class of a log if lower limit of the cutting window is below lower length class limit:</b> 1...var116/1...var111
135	4	Upper <b>length</b> limit for "cutting window"/price matrix/tree species. <b>It does not affect length classification of a log. It can not be above lower length class limit (132_t1) nor above lower length limit for the "cutting window" (135_t3), of the next length class:</b> 1...var116/1...var111

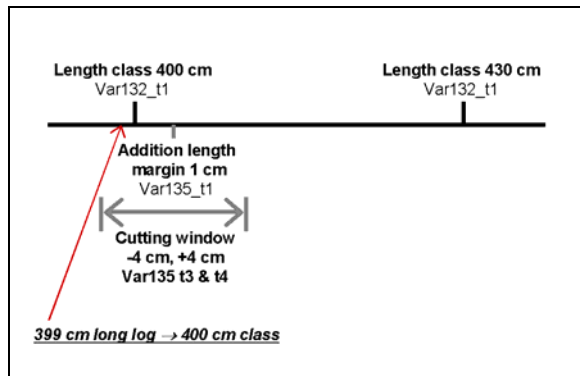


Figure 1. A log with a length of 399 cm will be registered in the 400 cm class

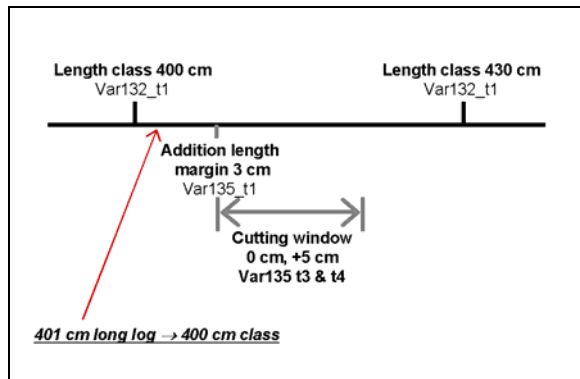


Figure 2. A log with a length of 401 cm will be registered in the 400 cm class while a log with a length of 399 cm will not be registered in the 400 cm class

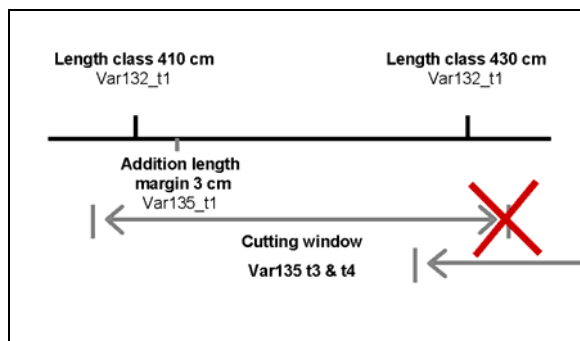


Figure 3. This is not a correctly defined "cutting window" since the upper length limit is above the lower length class limit or above the lower length limit for the "cutting window", of the next length class.

## 8 Variable 190 – MAXLOG

It was decided to introduce this variable in 1999 (StandForD minutes, 1999-10-04, §2). Variable 190 is closely related with variable 196, OVERPROD, which contains information about what action is to be taken when production target/limitation of variable 190 is reached. There are three different codes/actions for variable 196: 0 = No action (default), 1 = production is stopped, 2 = production is stopped and the price is set to zero.

### 8.1 Type 1

Type of limitation (*maximum allowed*) per price matrix and tree species.

1... var116/ 1... var111

Codes for limitation of production

0 = No limitation

1 = Total number of logs, quantity per price matrix

2 = Total volume of logs, m<sup>3</sup> per price matrix

3 = Number per diameter class per price matrix

4 = Volume per diameter class per price matrix

5 = Number per length class per price matrix

6 = Volume per length class per price matrix

- 7 = Number per cell (log size class) per price matrix and tree species
- 8 = Volume per cell (log size class) per price matrix and tree species

## 8.2 Type 2

Type 2 was finally defined in 2000 (StandForD minutes, 2000-04-10, §2)  
 Limit according to type 1 (*var 190*), specified for all cells in all price matrixes for all tree species.

1... var117/ 1... var118/ 1... var116/ 1... var111

Allowed special values:

0 = No limit

-1 = Forbidden to buck the log, *including manual bucking*

-2 = The log is excluded from apportionment bucking, just bucking according to value

-3 = *Only allowed to buck logs manually,*

As cell with a value of -1 totally forbids a log dimension to be bucked while a value of -2 indicates a division between apportionment bucking and bucking to value in the LOGMAX-matrix.

Only bucking to value is allowed when the value is set to -2. This alternative means that there are no requirements concerning the number or volume of logs for these cells. If the optimum alternative, when bucking to value, is a log with value -2 it will be bucked. However, if there is an apportionment bucking alternative with a price that is within the allowed deviation and there is a shortage for this log dimension this alternative should be selected.

## 8.3 Example

In the following example there is just one price matrix. Codes 1, 3, 5 and 7 are exemplified, these are basically comparable to codes 2, 4, 6 and 8.

### 8.3.1 Code = 1

Total number, quantity per price matrix

190 1 1~

190 2 50 0 0 0 0 300 0 0 -1~

	160	180	200
400	50	0	0
440	0	0	0
500	0	300	-1

The limit is the sum of all the cells with value  $\geq 0$ .

Total limit for matrix is 350

Logs with diameter 200 mm and length 500 cm are totally forbidden to buck.

### 8.3.2 Code = 3

Number per diameter class per price matrix

190 1 3~

190 2 50 0 0 0 0 300 0 0 -1~

	160	180	200

400	50	0	0
440	0	0	0
500	0	300	-2

The limits are the sums of all cells per diameter class with values  $\geq 0$ .

160 mm: Limit=50

180 mm: Limit=300

200 mm: No limit, the length 500 is excluded from apportionment bucking, only bucking according to value is allowed.

### 8.3.3 Code = 5

Number per length class per price matrix

190 1 5~

190 2 50 0 0 0 0 300 0 0 -1~

	160	180	200
400	50	0	0
440	0	0	0
500	0	300	-1

The limits are the sums of all cells per length class with values  $\geq 0$ .

400 cm: Limit=50

440 cm: No limit

500 cm: Limit=300 , the diameter 200 is forbidden

### 8.3.4 Code = 7

7= Number per cell (log size class) per price matrix and tree species

190 1 7~

190 2 50 0 0 0 0 300 0 0 -1~

	160	180	200
400	50	0	0
440	0	0	0
500	0	300	-1

There are three limits in this case:

No more than 50 logs with diameter 160 mm and length 400 cm.

No more than 300 logs with diameter 180 mm and length 500 cm.

Logs with diameter 200 mm and length 500 cm are forbidden to buck.

## 9 Variable 196 – OVERPROD

It was decided to introduce this variable in 1999 (StandForD minutes, 1999-10-04, §2, minutes only in Swedish).

### 9.1 Type 1

Action if production target is fulfilled (overproduction according to var190):

1..var116/1..var111

0 = No action (default)

1 = production is stopped

2 = production is stopped and the price is set to zero

Variable 190 will not have any effect if variable 196 is missing. Variable 196 is, in other words, mandatory if variable 190 is to be used.

There are today three codes dimensioned over 1..var116/1..var111, that is per price matrix and species:

0=no action

1=production is stopped

2= production is stopped and price is set to zero

Variable 190 will not have any effect on the bucking, if var196 has code 0.

However this code might be of interest if the same limitations are to be used intermittently, for example during May, June And August but not during July.

It will then be quite easy to shut off the limitations in July by setting

Var196T1=0.

Code 1 means that the production is stopped so that only manual cuts (forced cuts) can be done for a certain length/diameter/price matrix, whose limit is reached.

Code in Var190 and Var196 can only have an effect on price matrixes. The apportionment matrix (Var191) is not to be affected by Var190 and Var196.

### 9.2 Example

We have an APT-file with five price matrixes, each with two diameter classes and two length classes. We have agreed on a certain distribution with a sawmill for the coming months. However, with a short notice the sawmill gets an order that means that they will urgently need logs with a diameter of 180 mm and a length of 535 cm. A total of 1000 logs are needed within two weeks. A special assortment with its own price matrix (number 6) is defined. The limit on this assortment is set in variable 190.

The following data can then be found in the APT-file:

...~190 1 0 0 0 0 1~...(no limit in price matrixes number 1, 2, 3, 4 or 5, a limit on the total number of logs in price matrix number 6)

...~190 2 **0000** 0 0 0 0 **0000** 0 0 0 0 **0000** 1000~ ....(every second price matrix is marked with bold text for simpler reading)

No limits in price matrix 1,2,3,4 or 5. One cell in price matrix number 6 with the limit of 1000 logs.

...~196 1 0 0 0 0 0 1~.....(Production in matrix 6 is stopped when the limit in var190 is reached while there are no limitations for matrixes 1, 2, 3, 4 or 5)

## 10 Variable 197 – FROMMATRIX

It was decided to introduce this variable in 1999 (StandForD minutes, 1999-10-04, §2, only in Swedish).

Variable 197 defines which assortments that are permitted to cut as the first log as an alternative to the optimal bucking for the first log in apportionment bucking, in order to full fill the demands in the apportionment table.

The reason for using apportionment bucking is to in a simple way get a large proportion of logs with from a specified assortment within specified diameters och lengths. Thus when using apportionment bucking, it should by default be permitted to cut the first log from any assortment as an alternative to the optimal bucking.

However this might mean that certain assortments loose a very large part of their volume. The objective of variable 197 is in other words to “protect” an assortment from “theft” which is caused by apportionment bucking.

VarNo	Name	Type	DataType	Unit	Description
197	FROMMATRIX	1	Integer	Code	<p>From matrix: Code 0 = No, 1 = Yes            Price matrix for optimal alternative/            price matrix for alternative/species:            1...Var116/1...var116/1...var111</p> <p>From matrix: A matrix which defines which assortments (price matrixes) are allowed as an alternative to the optimal one for the first log in apportionment bucking in order to fulfill the demands in the apportionment table (var 191).</p>

An example:

~197 1 **101101** 0 1 0 0 0 0 **001000** 0 0 0 1 0 0 **000010** 0 0 0 0 1 ~...  
 (every second price matrix is marked with bold text to simplify reading)

This corresponds to the following matrix where it is permitted to cut a log according to price matrix number 3, 4 and 6 if the optimal alternative is a log belonging to price matrix number 1. However it is not permitted to cut a log according to price matrix 2 or 5.

In the case of, for example, assortment 3 being the optimal assortment it is not permitted to cut a log from any other assortment.

Price matrix no	Optimal Alternatives					
	1	2	3	4	5	6
1	X					
2		X				
3	X		X			
4	X			X		
5					x	
6	X					X

## 11 Variables 221 - NUMSTMS and 290 - NUMLOGS

An important issue is whether to include summed variables in the pri-file. Summed variables mean variables that can be calculated by using already existing variables, for example “Total volume =  $\Sigma$  Log volumes”. This kind of variable might be of interest when checking the results, however they do not increase the amount of information in the file.

It was therefore needed to add some new variables for checking the total results. This is especially true if we both cumulative and non-cumulative pri-files are used. The following two variables were therefore added (2003-04-08) to the standard with priority 1 in pri-files:

Var	Name	Type	Data type	Unit	Description
221	NUMSTEMS	2	integer	integer	Total number of trees harvested on a site after the start date (STARTDATE, var16_t4). Never reset during harvesting at a specific site. Used for checking that all files have been included when calculating the total production of a site.
290	NUMLOGS	2	integer	integer	Total number of logs harvested on a site after the start date (STARTDATE, var16_t4). Never reset during harvesting at a specific site. Used for checking that all files have been included when calculating the total production of a site.

## 12 Variable 241 - TOTMERCVOL and Variable 249 – TOTALVOL

These two variables are both used for registration of the total harvested volume in harvesters.

## 12.1 Historical background

The wood volume harvested is normally registered in a prd-file according to price matrixes defined in the apt-file. Logs cut of other dimensions than defined in the price lists have previously been registered in variable 244 OUTSPECVOL (earlier Swedish name STOLP) or in variable 243 BUTTVOL (earlier Swedish name LUMP).

At the StanForD meeting 2003-10-28 a special matrix for unclassified logs was decided (var244, next section). This matrix has wide length and diameter ranges to be able to catch up all dimensions which are not defined in the ordinary price matrixes. In connection with this a problem was been raised concerning the inclusion of the actual volumes in var243 or var244 (for unclassified logs) in the total volume (variable 241).

For the future, particularly when harvester measurement will be the basis for wood purchase, it is necessary to know both the total merchantable volume registered in the ordinary price matrixes defined by the apt-file and the total volume passing through the harvester. This is also important when pricing wood according to stem volume. Knowing both types of volumes it is up to the seller and the buyer to decide which volume to fix the price of.

## 12.2 Description

Therefore it was decided (2002-04-08) to change the name and improve the definition of variable 241 as well as creating a new variable 249 TOTALVOL. All according to the table:

Var	Name	Type	Data type	Unit	Description
<u>241</u>	<u>TOTMERCVOL</u>	<u>All types (1-6)</u>	<u>Integer</u>	<u>M3</u>	<u>Total merchantable solid volume (excluding price matrixes tagged with var136 code 4, as well as var243 and var244)</u>
<u>249</u>	<u>TOTALVOL</u>	<u>All types (1-6)</u>	<u>Integer</u>	<u>m3 ***</u>	<u>Total solid volume under/over bark (including price matrixes tagged with var136 code 4, as well as var243 and var244) .....</u>

TOTMERCVOL includes all price matrixes involved in the bucking calculation. TOTALVOLU includes TOTMERCVOL, the unclassified wood matrix or volumes registered in variable 243 and 244.

Variable 504 VOLDBHCL must include, as it says, the total volume, comparable with variable 249.

## 13 Variable 244 – OUTSPECVOL

Historically over or under sized (that is unclassified) logs were stored in variable var243 (BUTTVOL) or var244 (OUTSPECVOL). A major problem with these old variables was that we do not have any knowledge concerning the



diameter and length distribution of the unclassified logs. It is, for example, very difficult to analyze whether the unclassified volume is caused by incorrect handling by the operator or if it is due to errors in the apt-file. This issue was discussed continuously at StanForD-meetings since April 2001.

The objective of the unclassified matrix is to register all dimensions of logs (manually) produced where the quality and dimensions (too short or too long or under- or oversized) are not found by the computer in any of the normal commercial matrixes included in the apt-file.

It was therefore decided (2003-10-28) to introduce the following new variable with priority 1 in prd-files:

Var	Name	Type	Data type	Unit	Description
244	OUT-SPEC-VOL	5	Int	Code	Diameter class type in "Unclassified matrix": 1...var111 Codes: 0=Diameter classes are under bark, 1=Diameter classes are on bark.
	OUT-SPEC-VOL	6	Long int	No	Number of "unclassified logs" per fixed diameter and length classes, as well as species: 1...13/ 1...12/ 1...var111 Diameter classes interpreted according to var244_t5 (under bark or on bark) Fixed diameter classes:0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600+ Fixed length classes:0, 10, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900+
	OUT-SPEC-VOL	7	Long int	m3 sub	Volume of "unclassified logs" solid under bark per fixed diameter and length classes, as well as species: 1...13/ 1...12/ 1...var111 Diameter classes interpreted according to var244_t5 (under bark or on bark) Fixed diameter classes:0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600+ Fixed length classes:0, 10, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900+
	OUT-SPEC-VOL	8	Long int	m3 sob	Volume of "unclassified logs" solid on bark per fixed diameter and length classes, as well as species: 1...13/ 1...12/ 1...var111 Diameter classes interpreted according to var244_t5 (under bark or on bark) Fixed diameter classes:0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600+ Fixed length classes:0, 10, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900+

Figure illustrating how the fixed matrix for storing unclassified logs will look. The figure also illustrates examples of how location of logs in different parts of the matrix (A, B or C) may be analysed.

Top diameter, cm	0	5	10	15	20	25	30	35	40	45	50	55	60+
Length, cm	Large number of butt end adjustments												
0	A A A A A A A												
10													
50	B B B B B B B												
100	B B B B B B B												
200													
300													
400													
500	C C												
600	C C												
700	C C												
800													
900+													

Annotations in the figure:

- Green oval around row 0: Large number of butt end adjustments
- Red oval around rows 50 and 100: Cut short pieces of the stem due to root rot
- Blue oval around rows 500, 600, 700: Poor control of length measuring or large number of top breakages

## 14 Variable 503 - NUMSTEMDB

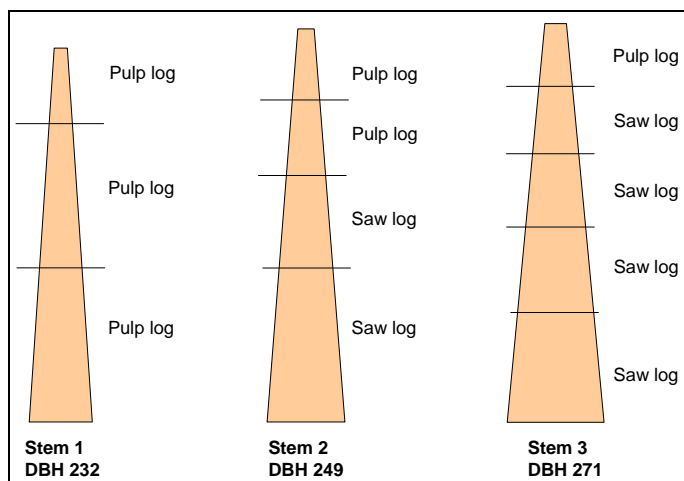
Variables 500-504 in StanForD are used for recording the DBH-distribution at a logging site. These variables are also used for recording the volumes per assortment and DBH-class. There are various possibilities to use this type of data, one example is to analyze the average stem size per DBH-class. Another application is to look at certain assortments percentage of certain DBH-classes. This may help in locating certain operations, for example harvesting of certain log qualities, to certain areas and stand types.

The idea with var503\_t2 (stems per price matrix) is to be able to analyse the number of trees from which a certain assortment has been bucked. It is for example of interest in certain situations to know the number of trees that has been infected by root rot as well as the average “rotten” volume of timber per tree. This is based on the fact that it is not uncommon to register these logs in a specific matrix. The variable was implemented as part of the work to develop tree pricing since it was then of interest to study both the frequency of trees with root rot as well as the frequencies of stems with other qualities (for ex quality 1, that is the “best” quality).

Note: Because one stem can contain logs of many assortments, total number of stems in var503\_t2 can not be compared with the total number of stems registered in for example var503\_t1 or var222\_t1.

### Example describing implementation of var503.

The figure illustrates an example of how three different stems (same species) were bucked into twelve logs.



The resulting data as it is to be registered in the prd-file is described in the table. Only one species, two price matrixes and three DBH-classes are used.

Var	Type	Name	Description	Data from example
111	1	NUMTREESPC	Number of tree species	1
116	1	NUMASST	Number of price matrices/tree species: 1...var111	2
121	1	ASSTDESCR	Description of assortment/price matrix/tree species: 1...var116/1...var111	"SawLog" "PulpLog"
501	1	NUMDBHCL	Number of dbh classes/tree species: 1...var111	3
502	1	LOWDBHBRK	Lower DBH limit on bark/dbh class/tree species: 1...var501/1...var111	220 240 260
503	1	NUMSTEMDB	Number of stems/DBH class/tree species: 1...var501/1...var111	1 1 1
	2		Number of stems/DBH class/price matrix/tree species: 1...var501/1...var116/1...var111	0 1 1 1 1 1
	3		Number of logs, including unclassified logs/DBH-class/tree species: 1...var501/1..var111	3 4 5
	4		Number of logs/DBH class/price matrix/tree species: 1..var501/1...var116/1..var111	0 2 4 3 2 1

## 15 Variable 523 - COORDSTM

In order to store coordinates in stm-files variable 520 (COORDREF), 521 (COORDTYPE) and 523 (COORDSTM) has to be implemented. Variable 522 (COORDSTART) has to be included if the coordinates in var523 are stored as a difference from the starting point.

It was decided to include coordinate variables in the pri-file at meeting in October 2002 (minutes 2002-10-28).

There are many advantages to collect coordinates. The simplest case is to monitor the position of the harvester and connect it to each tree harvested at that spot.

To save data space it seems advantageous to register a start point (e.g. at the landing) at the logging site and after that only store the differences from that point.

The following new variables and log codes were decided.

Var	Name	Type	Data type	Unit	Description
520	COORD-REF	1	integer	Code	Position of coordinate registration in file: 1= Base machine position 2= Crane tip position when felling the tree 3 = Crane tip position when processing the tree
521	COORD-TYPE	1	integer	Code	1=coordinates stored as a difference from the starting coordinates (COORDSTART var522) 2=absolute coordinates are stored in the file.
		2	integer	Coder	Coordinate system used in file: 1=WGS84 (Default)
522	COORD-START	1	integer	0.00001 degrees	Latitude when file is created (absolute value), registered according to var521_t2 and var520_t1.
		2	integer	Code	1=North, 2=South
		3	integer	0.00001 degrees	Longitude when file is created, (absolute value), registered according to var521_t2 and var520_t1.
		4	integer	Code	1=East, 2=West
		5	integer	Meters	Altitude, height above sea level, registered according to var521_t2 and var520_t1.
		6	string	yyyy-mmdd	Date and time

In order to make it possible to record the coordinates of individual trees in stm-files a new coordinate variable was decided (2003-08-08). It is structured as tree codes 760-764 (var266) in pri-files.

Variable STMCOORD (var523) with priority 3 in stm-files is defined according to the following:

Var	Name	Type	Data type	Unit	Description
523	COORD-STM	1	integer	0.00001 degrees	Latitude of stem, registered according to var521_t1, var521_t2 and var520_t1
		2	integer	Code	1=North, 2=South The code in var522_t2 is valid for all stems if this code is excluded
		3	integer	0.00001 degrees	Longitude of stem, (absolute value), registered according to var521_t1, var521_t2 and var520_t1
		4	integer	Code	1=East, 2=West The code in var522_t4 is valid for all stem if this code is excluded
		5	integer	Meters	Altitude of stem, height above sea level, registered according to var521_t1, var521_t2 and var520_t1
		6	string	yyyy-mmdd-hhmmss	Date and time when coordinate in var523 was recorded

## 16 Revisions of document

All major (not spelling/grammar) changes are to be described as well as date and the person responsible.

No.	Date	By:	Description
1	2002-02-27	JOAR	Variable 197 added to document
2	2002-04-09	JOAR	Variable 241, 249
3	2003-02-07	JOAR	Variable 132, 135, 22, 60
4	2003-04-30	JOAR	Variable 520-523, 221, 290
5	2004-02-10	JOAR	Variable 244, 503, 241, 249
6	2006-05-29	JOAR	Variable 105

### Abbreviations:

JOAR = John Arlinger

# 17Index

<b>B</b>		<b>S</b>	
BARKPAR .....	6	STARTDATE .....	2, 4
<b>C</b>		<b>T</b>	
COORDSTM.....	17	TOTALVOL.....	13
Cutting window .....	6	TOTMERCVOL.....	13
<b>E</b>		<b>U</b>	
ENDDATE .....	2, 4	unclassified logs.....	15
<b>F</b>		<b>V</b>	
FILEORDER.....	1	var105 .....	5
FROMMATRIX.....	12	var11 .....	4
<b>G</b>		var112 .....	6
GRADESYS.....	5	var113 .....	6
<b>I</b>		var12 .....	4
Introduction .....	1	var132 .....	6
<b>L</b>		var135 .....	7
LWRLNGTHLMT .....	6	var16 .....	4
<b>M</b>		var17 .....	2, 4
MASS .....	4	var190 .....	8
MAXLOG .....	8	var196 .....	11
<b>N</b>		var197 .....	12
NUMLOGS .....	13	var22 .....	1
NUMSTEMDB .....	16	var221 .....	3, 13
NUMSTMS .....	13	var237 .....	4
<b>O</b>		var241 .....	13
OUTSPECVOL.....	14	var244 .....	14
OVERPROD .....	11	var249 .....	13
OVERSIZE .....	7	var290 .....	3, 13
<b>R</b>		var5 .....	1
RESETDATE .....	2, 4	var503 .....	16
Revisions of document .....	19	var504 .....	14
		var520 .....	17
		var521 .....	17
		var522 .....	17
		var523 .....	17
		var60 .....	4
		VARIANT .....	1
		VOLDBHCL.....	14
		<b>W</b>	
		WEIGHTCAL.....	4

Ämnesord: