

Minutes from the meeting with the StanForD-group 2003-04-08

Place: Skogforsk, Uppsala, Sweden

Present: Jari Korhonen, Ponsse
Jyrki Partanen, Ponsse
Anders Emmoth, Ponsse
Ola Söderlund, Partek
Joakim Nygren, Haglöf
Anders Hansson, SCA
Jaakko Mäkinen, Mitron
Tapio Räsänen, Metsäteho
Samuli Hoja, Metsäteho
Dan Norman, Caterpillar
Rolf Volungholen, Caterpillar
Nicklas Götz, CC Systems
Ilari Virtanen, Timberjack
Mika J Talola, Timberjack
Leif Orth, Södra
Mats Holknekt, Korsnäs
Tor Marntell, VMR (after lunch)

From Skogforsk: John Arlinger, Secretary
Jan Sondell, Chairman
Johan J. Möller

§1 Opening of the meeting

Jan Sondell opened the meeting. The proposed agenda was accepted.

§2 Clarifications of variables

FILEORDER, var22

A new variable, FILEORDER, was introduced at the last meeting (2002-10-28). John Arlinger presented the background to FILEORDER and the ideas concerning how to implement the variable (appendix 2).

It was decided to change the priority from 1 to 3 in prd-files (keeping it as priority 1 in pri-files) and to update the description according to the following (new text in bold):

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 The file is considered to be cumulative if variable is missing. Prd-files are normally cumulative and pri-files are normally non-cumulative.

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 file while a non-cumulative file is reset after reporting.

The meeting decided that if FILEORDER is missing the file is considered to be cumulative.

It was also decided that the recommended default should be that:

- prd-files are stored as cumulative files
- pri-files are stored as non-cumulative files

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 cumulative 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	2600	2600	4	No	3	01-01-05

To calculate the total production, files from day 1, 4 and 3 (the last file with filorder 2) are all summed. This means that the last file with a specific FILEORDER (in this case the second file with FILEORDER 2) is always used when calculating the total sum.

It was decided (see §4) to add a sum variable to use as a checksum in order to be certain that no file is missing in a case as described above.

The issue of whether we need one more variable to express if the production data in a file is cumulated or not was also raised at the meeting. The meeting decided that it is not needed since FILEORDER is only updated when no old data is stored in the file, meaning that they are not cumulated. If a person has

one file with FILEORDER equal to 1 and another file with FILEORDER equal to 2, both files have to be summed in order to get the total production. On the other hand if both files have FILEORDER equal to 1 (or if FILEORDER is missing) you know that they are cumulative files. A cumulative file is defined as a file which is not reset between reporting.

It was also decided at the meeting to improve the description of variable RESETDATE (var11_t4) according to the following (new text in bold):

Var	Name	Type	Data type	Description
11	RESETDATE	4	integer	Time and date when file was last reset (mandatory), normally at the same time as production is started

Definition of checksums

Partek, has found a problem with the declaration of variable **ch** in the code example *Funs.cpp* (can be found at www.skogfork.se) This example is used for describing how to calculate the check sums. The example is written in C-code (see below).

```
static void UpdateChecksums(SChecksums& chsum, char ch)
{
    unsigned short q, crc;
    chsum.Mchecksum1 = (short)(chsum.Mchecksum1 + (short)(ch));
    crc = chsum.Mchecksum3;
    q = (short)((crc ^ (short)(ch)) & 15);
    crc = (crc >> 4) ^ (short)(q * 4225);
    q = (short)((crc ^ (short)((short)(ch) / 16)) & 15);
    crc = (crc >> 4) ^ (short)(q * 4225);
    chsum.Mchecksum3 = crc;
}
```

Variable **ch** is declared as a **char** which depending on environment/compiler can be interpreted as signed or unsigned, meaning that the value of the variable can be either -128..127 or 0..255. This has the consequence that the calculated checksum could get different values in the case a file contains characters above 127.

It was therefore decided that variable **ch** is defined as an **unsigned char** (0..255) as in the example below:

```
static void UpdateChecksums(SChecksums& chsum, unsigned char ch)
```

INITGRAD, var144

Variable INITGRAD (var144) has been implemented in a binary format in some applications. Traditionally this variable has not been stored as a binary value. Stefan Falk therefore suggested that the description is updated so that it is clearly stated that this variable is not to be stored in a binary format.

The proposal was rejected since most manufacturers use a binary format for var144.

Comment from secretary: The problem remains and we probably should discuss the issue again with representatives present from manufacturers implementing the variable in a non

binary format. The administrative program WinApt handles the problem, storing INITGRAD both as binary and non binary depending on type of machine, while the administrative program SilviA always store the variable in a binary format. The problem would lessen, at least in Sweden, if SilviA was modified so that it functions in the same way as WinApt.

§3 New variables

SITEINFO, var23_t1

There is no variable in the standard for expressing the logging type (clearcutting, first thinning, later thinning, removal of seed trees etc.) of the site.

To make it possible to register logging method it was decided that a new variable is introduced with priority 3 in prd- and pri-files:

Var	Name	Type	Data type	Unit	Description
23	SITEINFO	1	string	code/text	Harvesting method, the code is not standardized in StanForD

The meeting decided that Skogforsk and Metsäteho is to study the possibility of introducing a standardized list of logging type codes.

Comments added after the meeting by the secretary: It would probably be suitable to store the information in var21_t5 (certified site or not) in var23, however since var21_t5 already has been implemented this probably is of limited interest.

ASSTDSCR, var121_t3

An additional variable for identifying price matrixes is needed in Sweden as originally suggested by Leif Orth, Södra. This is due to the fact that, in Sweden, var121_t2 is based on a standardized code that does not change over time. This standardized code is also in some cases identical for several price matrixes.

In order to keep track of what version (for example date of creation) of price matrix, the following new variable was decided with priority 2 in apt-, prd- and pri-files:

Var no	Name	Type	Data type	Unit	Description
121	ASSTDSCR	3	String	Text	Additional identity description of price matrix /price matrix/ species: 1...var116/1...var111

VARIANT, var5_t3

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 discussed, 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 StanForD variables in present file was carried out. Version of StanForD is the meeting date when last update of standard was decided e.g. 20030408.

The date in var5_t3 is identical with the date of the StanForD-meeting (in this case 2003-04-08). This date is presently used for dating the minutes as well as all variable lists and the appendix.

Comments added after the meeting by the secretary: It was, after a discussion with Partek, decided by the secretary to postpone the implementation of this variable to a later meeting due to its rather vague description. It can, as defined above, be interpreted in several different ways.

Coordinates in stm-file, var523

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

It was decided to include a new variable, COORDSTM (var523), with priority 3 in stm-files according to the following description:

Var	Name	Type	Data type	Unit	Description
523	COORD	1	integer	0.00001 degrees	Latitude of stem, registered according to var521_t1, var521_t2 and var520_t1. When var521_t1 = 1 this variable (var523_t1) is recorded as the difference from var522_t1.
		2	integer	Code	1=North, 2=South The code in var522_t2 is valid for all coordinates 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. When var521_t1 = 1 this variable (var523_t3) is recorded as the difference from var522_t3.
		4	integer	Code	1=East, 2=West The code in var522_t4 is valid for all coordinates 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. When var521_t1 = 1 this variable (var523_t5) is recorded as the difference from var522_t5.
		6	string	yyyy-mmdd-hhmmss	Date and time when coordinates in var523 were recorded
		7	integer	code	Code for type of object with position according to coordinates in var523, point of measurement is according to var520_t1: 1 = harvested stem (in stm-file)
		8	string	text	Free descriptive text

It was also decided that variables 520 (COORDREF), 521 (COORDTYPE) and 522 (COORDSTART) should be included in the stm-file with priority 3.

Observe that 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 (var522).

Var523 is to be repeated for each stem in a stem-file where more than one stem is stored.

Below are two examples illustrating how var522 and var523 are implemented in stm-files. The format is slightly changed in order to make it more readable. Observe that a large number of important variables have been excluded and that only the first two diameter values are stored in STEMEDIA (var273_t3):

Example 1. Coordinates are stored in var523 as a difference from the starting point in var522:

```

1 2 stm ~
520 1 1 ~
521 1 1 ~
522 1 5996423 ~
522 3 1867654 ~
522 5 35 ~

521 2 1 ~
522 2 1 ~
522 4 1 ~
522 6 20030522145510 ~

110 1 1 ~
270 1 1 ~
272 1 225 ~
523 1 134 ~
523 5 2 ~
523 7 1 ~

269 2 10 ~
271 1 0 ~
273 3 330 2 2 .....~
523 3 69 ~
523 6 20030522145749 ~
523 8 Rel_Test_1 ~

110 2 1 ~
270 1 2 ~
272 1 210 ~
523 1 94 ~
523 5 -1 ~
523 7 1 ~

269 2 10 ~
271 1 0 ~
273 3 293 3~ 2 .....
523 3 101 ~
523 6 20030522145955 ~
523 8 Rel_Test_2 ~

```

Example 2. Coordinates are stored in var523 as absolute values:

```

1 2 stm ~
520 1 1 ~
521 1 1 ~
522 1 5996423 ~
522 3 1867654 ~
522 5 35 ~

521 2 1 ~
522 2 1 ~
522 4 1 ~
522 6 20030522145510 ~

110 1 1 ~
270 1 1 ~
272 1 225 ~
523 1 5996557 ~
523 5 37 ~
523 7 1 ~

269 2 10 ~
271 1 0 ~
273 3 330 2 2 ..... ~
523 3 1867723 ~
523 6 20030522145749 ~
523 8 Abs_Test_1 ~

110 2 1 ~
270 1 2 ~
272 1 210 ~
523 1 5996517 ~
523 5 34 ~
523 7 1 ~

269 2 10 ~
271 1 0 ~
273 3 293 3 2 ..... ~
523 3 1867755 ~
523 6 20030522145955 ~
523 8 Abs_Test_2 ~

```

§4 Pri-file

Updating and adding variables in pri-file

The following updates are largely based on comments and questions from the Finnish pre-StanForD-meeting held in February, 2003.

New variable

The following updated description of a code in variable 266 (TREECODE) was decided (new text in bold):

Tree code	Old description	New description
740	DBH (on bark), mm	DBH (on bark) measured at a height according to var500_t1 , mm

New log code

The following new log code (LOGCODE) was decided:

Log code	Description
300	Forced cross-cut (break) at small end, code according to var300.

Clarification

At the Finnish meeting it was discussed whether to change the unit of variable 258 (DISTANCE) from km to meter (see table below). It was also discussed whether to change or improve the description. We could either change the unit of the present variables or introduce two new variables (type 3 and 4) that are identical with the present types except for the unit. An improved description of var258, based on Partek's original proposal, is suggested with the minutes (in bold style).

Var no	Type	Name	Unit	Description	Prio
258	1	DISTANCE	km	Covered distance, variable is reset together with production	3
258	2	DISTANCE	km	Covered distance/operator, variable is reset together with production: 1...var211	3

Comments added after the meeting by the secretary: Since the variable is already implemented by Partek it is not possible to change the unit. As it is possible to use it as a "1000-variable" where the decimal part is stored in var1259 it seems needless to add two new types. If this is still considered to be of vital interest it is suggested that the issue is discussed again at the meeting in October.

Updating priorities

It was decided that the following priorities in pri-file are updated:

Var no	Type	Name	Description	Old prio.	New prio.
5	1, 2	VARIANT	Model variant for the bucking computer...	0	3
13	4	BUIDATE	Time and date when the current bucking file was created...	1	2
21	1	SITENO	Logging unit number	1	2
21	2	SITENO	Marking for cutting	3	2
21	3	SITENO	Compartment number	3	2
21	4	SITENO	Lot number	3	2
21	5	SITENO	Code to express if the logging site is certified or not	3	2
33	1	VENDOR	Vendor	0	3
40	1	NMLNGTHCA	Number of length calibrations	0	2
40	2	NMLNGTHCA	Number of length calibrations per tree species	0	3
43	1	NUMDIACAL	Number of diameter calibrations	0	2
43	2	NUMDIACAL	Number of diameter calibrations per tree species	0	3
121	2	ASSTDESCR	Code/price matrix/tree species:	2	1

122	1	NUMSTEMTY	Number of stem types/species	0	2
124	1	STEMTYP-DES	Description of stem type/stem type/species	0	2
124	2	STEMTYP-DES	Code for stem type/stem type /species. Finnish PMP-codes	0	2
200	1	OPTNLTXT	For inclusion of optional text	4	0
243	1	BUTTVOL	Volume of butt-offs/tree species (shorter than spec.)	3	0
250	1	SAWLGVOL	Sawlog volume/tree species	3	0
274	2	NUMGRADE-BR	By calculation automatically generated grade limits	3 / 0	0
275	3	HGHTGRAD-BRK	Height where stem diameter is equal to the diameter of the grade cylinder in var274T2	3 / 0	0
276	3	GRADE	Specification of qualities (from but end) generated automatically by calculation	3	0
504	3	VOLDBHCL	Total volume including unclassified assortments, sum of var504 should be equal to sum of var249, /DBH class/tree species: 1	3	0
504	4	VOLDBHCL	Volume/ DBH/price matrix/ species	3	0
504	5	VOLDBHCL	Volume, unit according to price matrix/DBH /price matrix/species	3	0

Observe that that the following variables have a priority other than 0 probably due to errors by the secretary: var200, var 243, var250, var274, var275, var276, var504

Variables for checking total production

An important issue is whether to include summed variables in the pri-file. Summed variables mean variables that can be calcuted by using already existing variables, for example “Total volume = Σ 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 is probably needed to add some new variables for checking the total results. This is especially true if we use both cumulative and non-cumulative pri-files. It was decided that the following two new variables are added 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.

It was also decided that the secretary is to look into a possible improvement of the descriptions of type 1 (var221 and 290) before next meeting.

New document describing pri-files

John Arlinger informed about a document describing pri-files that was sent out with the agenda. An updated version of this document will be sent out together with the minutes.

§5 Items for discussion and information

VARIANT, var5

Tapio Räsänen informed the meeting about the problem that different manufacturers have implemented variable 5, type 1 and 2 differently.

UNITAREA, var670

Tapio Räsänen informed the meeting about a Finnish discussion concerning the fact that there is no variable in the standard for expressing the area of a logging unit. Var670 is reserved for inventory data. The unit of the area should not be m², perhaps 1/10 of a hectare would be a more suitable unit. The meeting decided that the issue of adding an area variable should be discussed at the next meeting.

Production file for forwarder

John Arlinger informed about ideas concerning how to implement a new file format for reporting forwarder production (appendix 1). A comment from the meeting was that the status of each landing per assortment must be included, for example whether it is empty or full. It may also be of interest to include a description of the landing, for example if it is shaded or not as this affects how quickly logs lose their moisture content.

The meeting decided that Skogforsk may continue the work and to present a proposal at the next meeting.

PRICECAT, var161_t1

The present description of price category 10 (var161_t1) is as follows:

Code 10. Name of category: M3sm

Price/m3. Solid volume, diameter (mm) measured at midpoint for calculating the volume, price due to small-end diameter (mm).

It was decided (as suggested by Stefan Falk, Rottne) that the name for code 10, PRICECAT (var161_t1), is to be changed from M3sm to M3smto

§ 6 Control and calibration of harvester

Johan J. Möller informed the meeting about the ongoing project concerning how to implement a certification system for harvester measurements.

The meeting decided to introduce the following new variables with priority 3 in stm- and ktr-files:

Var	Name	Type	Data type	Unit	Description
270	STEM- NUM	2	integer	integer	Unique number of sampled stem, used for identifying control stems. Should be 0 (or excluded) if not sampled for control.
306	LOGID	2	integer	integer	Unique number of sampled log, used for identifying control logs. Should be 0 (or excluded) if not sampled for control.

§ 7 Recommendations concerning unclassified logs

The meeting decided that the matrix for unclassified logs should be “hard coded” in var244 (OKLASS/UNSPEC) and that a proposal will be presented by Skogforsk at the next meeting.

§ 8 Standard for communication between machine control systems and other applications.

John informed about the ongoing work. The new working group organized by Bengt-Arne Walldén, Partek had a meeting in January. All participants agreed on the needs for standardizing the communication between machine control systems and other applications (GIS-applications). Unfortunately the response from the participating companies after the meeting was very limited. This meant that Partek decided to not continue their efforts with a large working group.

At the same time Sveaskog is writing on a specification describing their need for a standardization of harvesting directives, including geographical data. Thus

probably increasing the need for standardization of communication between machine control systems and GIS-applications.

Skogforsk foresees a probable need for standardization of both of these issues. Skogforsk will therefore strongly support any proposal concerning standardization of communication between machine control systems and other applications. The work at Sveaskog will also be followed closely and may be presented at the next meeting.

§9 Other StanForD issues

StanForD XML (Metsäteho)

StanForD-XML development project close to be finished at Metsäteho.

Translating programs and XML schemas are completed: PRD_to_XML.exe, STM_to_XML.exe, APT_to_XML.exe. A pri-schema will be developed according to the new description.

A final report is written in Finnish.

Collection of stm-files for databases (Metsäteho)

Tapio Räsänen informed the meeting about UPM-Kymmene's intention to start collecting stm-files from different areas in order to create a database for bucking simulations

The objective is to collect and transfer data, and then to store it in a database with no manual actions needed.

Requirements for files and variables have been described by UPM-Kymmene. The requirements have been given to machine manufacturers. The issue was discussed in Finnish StanForD meeting.

UPM-Kymmene's major requirements are:

- more than one stem should be stored per file
- not cumulative files
- interpretation of all information should be done without apt or prd (e.g. timber assortments)
- filtered diameter information
- all the necessary identification information of a site and machine must be included
- unclassified logs should be included in stm
- sampling should be possible

Colour marking of logs (Skogforsk)

Jan Sondell informed about problems with the fact that the colour used for marking logs fade quite quickly (in one case within two weeks).

Invitation to new studies (Skogforsk)

Johan Möller informed about a number of studies planned at Skogforsk together with Swedish users. Contact will be taken with the manufacturers.

Variables for monitoring (Skogforsk)

John Arlinger informed about the ongoing work and that a meeting for manufacturers is planned for the 8th of May.

§10 Next meeting

Next meeting will take place on Tuesday October 28th, 2003, at Skogforsk in Uppsala, starting at 10.00 a.m. Proposals for the meeting shall be the secretary by hand not later than September 15th 2003.

§11 End of meeting

Jan Sondell (chairman) closed the meeting.

Secretary

John Arlinger



Johan Möller
John Arlinger
2003-02-21

PRELIMINARY

A new StanForD-file for loads of timber – prl-file

Do we need a new type of production file for forwarders?

Background

An increasing number of manufacturers are installing PC-computers in their forwarders. This means that there are new possibilities to generate StanForD-files like drf- and prd-files from forwarders. Production files from forwarders are of great interest to many organisations/companies since information concerning available volumes of wood at the landing is very important for planning the wood flow from forest to industry.

There are two major problems with the present prd-file from a forwarder perspective:

1. Almost 50% of the variables in the prd-file are useless for forwarders
2. It is not possible to include production data concerning several landings on one site within the present structure without a major update of the file type.

A new type of production file is therefore suggested. This file should only be used by forwarders. A suggested name for the new type is prl-file (production load). This document contains minimum requirements on the information sent in the form of a production file from the forwarder. It also briefly describes a possible data structure. The suggestions below are to be seen as a basis for discussion and not as a final proposal to the StanForD-group.

Present prd-file

There are today in the standard 215 variables that can be included in the prd-file, excluding variables with priority 4. At least 85 (perhaps up to 102) of these variables can only be generated by harvesters and not by forwarders. This is due to the fact that most of these variables are based on information originating directly from the measuring devices of the harvester head. It is, for example, assumed that only a harvester, and not a forwarder, can keep track of stems (eg DBH) or stem types, number of logs, diameter and length of logs.

Approximately 25 variables in the prd-file originates directly from the apt-fil and 13 variables can only be generated by a forwarder. The forwarder variables have all been added to the standard quite recently.

Is it possible for a forwarder to keep track of assortments/price matrixes? Probably not in Sweden since logs belonging to different price matrixes often have the same dimensions and are delivered to the same customer in one batch. This is the reason why it was stated above that between 85 and 102 variables can be generated by a harvester. Look more at var242 and var249. How to classify these variables? Can they be generated by a forwarder?

Using other StanForD-files

To minimize registration errors as well as making the registration for the operator simple it is probably of interest to be able to utilise information in the

apt-file. It should, in the forwarder, be possible to read identities, price matrix codes, types of volumes etc. directly from the apt-file used by the harvester and use this information when recording data in the prl-file. It may also be of interest to read a pri/prd-file from the harvester on the same site in order to compare and possibly correct the transported volumes with the harvested volumes.

A new prl-file

Data that probably should be included in prl-file:

- Identity variables as in pri- and prd-files
- Volume (relevant units)
- Mass
- Density
- Production specified per:
 - load
 - operator
 - landing
 - species
 - price matrix
- Date and time of production/unloading
- Landing coordinates
- Calibration of scale

Structure of prl-file

The data could be stored using the same principles as the pri-file. That means that we store information concerning each individual load and assortment.

Table 2. Example

VarNo	Name	Type	Data							
YY5	NUMLOADCOD	1	7							
YY6	LOADCODE	1	X1	X2	X3	X4	X5	X6	X7	X8
YY6	LOADCODE	3	LoadNo	Assort-code	Buyer-code	Vol	Mass	CrdX	CrdY	Time
YY7	LOADDATA	1	1	301	135	9	7200	218	554	12:32:10
			1	222	133	3	2400	219	554	12:33:30
			2	701	133	13	10400	220	556	12:45:40
			3	301	135	12	9600	230	540	13:09:40

Table 3. StanForD-file example

YY5 1 7~ YY6 1 X1 X2 X3 X4 X5 X6 X7 X8~YY6 3 LoadNo Assort Buyer Vol Mass CrdX CrdY Time~ YY7 1 1 301 135 9 7200 218 554 12:32:10 1 222 133 3 2400 219 554 12:33:30 2 701 133 13 10400 220 556 12:45:40 3 301 135 12 9600 230 540 13:09:40~

Summary of presentation concerning implementation of FILEORDER

FILEORDER is used when dividing production from one site into several files!

Especially important when implementing pri-files.

Background for the examples following below:

- One site
- Start at 7.00, finish at 18.00 each day
- Three days of production
 - Day 1 - 25% is harvested
 - Day 2 - 35% is harvested
 - Day 3 - 40% is harvested
- VarSum = the sum of all logs/trees from a site, never reset (var221_t2 or var290_t2 as described in §4)

In the 1st example we only report production once. FILEORDER is equal to 1 and we have an ENDDATE. FILEORDER is not of any real interest:

1) Single file	
Prod data for 0-100%	
FILEORDER	1
STARTDATE	01, 7.00
RESETDATE	01, 7.00
SAVEDATE	03, 18.00
ENDDATE	03, 18.00
VarSum	100%

 SKOGFORSK

In the 2nd example we have multiple files, meaning that we report on a daily basis. Here we are using cumulative reporting as is normally done when reporting prd-files today. The same STARTDATE and RESETDATE in all files:

2) Multiple files, cumulative		
<i>Cumulative files = file <u>is not reset</u> after reporting, all data from a site at the time of reporting is stored in the file</i>		
<u>Prod data for 0-25%</u>	<u>Prod data for 0-60%</u>	<u>Prod data for 0-100%</u>
FILEORDER 1	FILEORDER 1	FILEORDER 1
STARTDATE 01, 7.00	STARTDATE 01, 7.00	STARTDATE 01, 7.00
RESETDATE 01, 7.00	RESETDATE 01, 7.00	RESETDATE 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%
Total sum = file 3 		

In the 3rd example we have multiple files, meaning that we report on a daily basis. Here we are using non-cumulative reporting as will probably be the normal procedure for pri-files. The same STARTDATE in all files, RESETDATE and SAVEDATE is updated in all files. We know that the third file is the final file since it has an ENDDATE. To calculate the total, file 1, 2 and 3 are summed.

3) Multiple files, non-cumulative		
<i>Not cumulative = file <u>is reset</u> after each reporting, meaning that data is reported only once.</i>		
<u>Prod data for 0-25%</u>	<u>Prod data for 25-60%</u>	<u>Prod data for 60-100%</u>
FILEORDER 1	FILEORDER 2	FILEORDER 3
STARTDATE 01, 7.00	STARTDATE 01, 7.00	STARTDATE 01, 7.00
RESETDATE 01, 7.00	RESETDATE 02, 7.00	RESETDATE 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%
Total sum = file 1+2+3 		

In the 4th example we have a mix, where the file is reset on the second day but not on the first. To calculate the total sum in this case take the last file with FILEORDER 1 and the last file which has FILEORDER 2 and an ENDDATE. It is possible to control that the correct files have been summed by comparing VarSum with the culculated total sum.

4) Multiple files, mix

<u>Prod data for 0-25%</u>	<u>Prod data for 0-60%</u>	<u>Prod data for 60-100%</u>
FILEORDER 1	FILEORDER 1	FILEORDER 2
STARTDATE 01, 7.00	STARTDATE 01, 7.00	STARTDATE 01, 7.00
RESETDATE 01, 7.00	RESETDATE 01, 7.00	RESETDATE 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%



Cumulative

Non-cumulative

Total sum = file 2+3

 SKOGFORSK