



Arlinger John

3 December 2020

Minutes StanForD

2020-12-02

Place:	Microsoft Teams (Uppsala), 2020-12-02, 09:00-16:00 CET
Participants:	John Arlinger, Secretary Johan J. Möller, Chairman Tapio Räsänen, Metsäteho Timo Melkas, Metsäteho Pasi Palonen, Parker Juha Korhonen, Technion Lars-Erik Andersson, Dasa Jonas Thorsson, Dasa Henrik Persson, Logmax Antti Korkiakoski, JohnDeere Erik Kindlund, JohnDeere Niclas Eriksson, Komatsu Jacek Nawrot, Tigercat Josh Hildebrand, Tigercat Heikki Komulainen, Ponsse Janne Pursiainen, Ponsse Asko Leivonmäki, Exertus (Logset) Ian Wilson, ForestPHD Andreas Barth, CGI Lars Henriksson, Biometria Mike O'Shea, Coiltte Jonas Gustafsson, Sveaskog Christer Ranvald, Sveaskog Patrik Anderchen, Södra Alexander Kaulen, KWF Marius Kopetzky, KWF Christophe Ginet, FCBA Jean Plamondon, FPIInnovation Jyrki Vainio, Trimble Bengt Sörvik, Otmeka

Decision summary:

- New messages for skidders and yarders accepted. Final decision to publish schemas will be made 2020-02-05.
- New API specification accepted, to be published as a draft during December 2020. Final decision to publish official specification will be made 2020-02-05.
- A very large number of updates to be added to beta-schema 4.0 (see minutes).
- The following issues to be discussed further at the next meeting:
 - Format restrictions on UserIDs
 - Unit of LogLength
 - Making SUBObjects mandatory
 - New Tracking structure
 - StemTypeCodes
 - Unclassified stems
 - new structure for DiameterVector in hpr and hqc.
 - New structure for storing geometries in ogi.
- Secretary to look at possibility to simplify mom regarding detailed down time reasons.
- Forwarder test planned for spring of 2021.

Action list (not updated since last meeting):

- The following guidelines should be followed when implementing LongTimberDefinition:
 - Must be possible to view and edit LongTimberDefinition in administrative tools.
 - Default value equal to DiameterMAXButt in administrative tools
 - LongLogButtMAX is never to be included in pin-file if value is 0 (it is optional).
 - Value 0 (zero) is to be interpreted as null.
- It is recommended to implement stem codes for both individual trees and groups of trees (so called “sticky registration of stem codes”). Whether the “sticky registration” is triggered by double click or any other operator interaction does not matter. It is an advantage if “sticky codes” is presented on, Screen if active.

- Do not mix products and deliveries in forwarder applications.
 - Focus on Deliveries in the UI.
 - Avoid using pin-files. It often creates a lot of confusion
- Map interface when creating Locations would be good in order to avoid incorrect coordinates.
- TextToMachine an urgent variable to be shown when creating new object in harvester and forwarder.
- The description of OtherMachineDownTimeStandardCode was updated: *Allowed values: "Waiting for repair", Trailer transportation", "Unproductive terrain work", "Other", "Default". "Default" used only in cases where operators have not registered any reason at all. "Default" not to be available in GUI and only generated by control system. "Default" to be used in system where down time reasons are disabled or there is a "time-out".*
- It is strongly recommended that the control system does not automatically activate a modified pin if it is not active at the present harvesting object
- All randomly selected stems (both rejected and not rejected) must be saved in an hqc-file including a possible reason for rejection. This file should be possible to send directly to the forest company (in Sweden to Biometria).
- Coordinates in production files should always be activated by default in all forest machines (especially important after software updates when coordinate registration is commonly lost).

§1 Opening of meeting and agenda

Chairman Johan J Möller opened the meeting.

Agenda accepted.

§2 Presentation of participants

Presentation of meeting participants.

§3 Information about decisions and discussions at previous meetings

The decisions made at the previous meetings were presented.

Meeting 2020-05-07:

- American volume unit Board feet: Only consider new volume types if there is a strong requirement to use it for optimization of bucking
- Study possibility a "log volume extension" in the hpr where manufacturer can store un-standardized volume types used for reporting
- Postpone the issue of controlling data in the hpr using the oin

Meeting 2020-08-17 (for members):

- Implement a new major version 4.0 in December 2021.
- Stay with version 3.6 at meeting 2020-12-02 with the possibility to introduce new yarder and skidder schemas within 3.6 at that meeting.

§4 StanForD API. New communication methods

The complete draft specification is enclosed in appendix 1. The main updates since last meeting 2020-05-07 are:

- Compressing large files (hpr) by using functionality built into http.
- New request “synchronization” in order to get information from server about last communication with machine,
- New method HEAD to determine file size prior to downloading of e.g. hpr.

It was decided to add the new specification to agenda for the meeting 2020-02-05 for a final decision to publish it as version 1.0. It was also decided to publish the draft specification at the Stanford web site.

A meeting was held between Skogforsk and Metsäteho in early November to discuss the future Data-APIs. The conclusions drawn at that meeting were:

- To be based on the existing schemas
- Focus on “filtering out” of optional and sensitive elements:
 - HarvestDate, StemCoordinates, BoomPosition, DiameterVector, Tracking
 - ContactInformation, (for specific definitions or all definitions?)

§5 New skidder and yarder schemas in version 3.6

The secretary presented the proposal for new skidder and yarder messages including instructions and production (appendix 2).

Observe that these new schemas (yoi, ypr, soi and spr) do not affect any of the existing version 3.6 schemas.

No objections raised.

It was decided to add this as a proposal to agenda for the meeting 2020-02-05 for a final decision to publish these new version 3.6 schemas in February.

§6 Proposals new major version 4.0 (Appendix 3)

DATA TYPE

New simple data types

It was decided (proposal at meeting by Henrik Persson) to study the possibility of implementing new simple types for handling commonly register data categories such as Lengths, Diameters, Volumes, Keys (numerical), PresentationOrders etc.

UserDefinedData

John Arlinger, Skogforsk (2014-04-08).

It was decided that tableName is given data type string and columnOrder is given data type nonNegativeInteger (comparable with rowOrder).

DeliveryPresentationOrder

Fredrik Tjernberg, Niclas Eriksson, Halvard Olderlund, Komatsu.

It was decided that DeliveryPresentationOrder is given data type positiveInteger as other PresentationOrders.

Error:

```
<xsd:element name="DeliveryPresentationOrder" minOccurs="0"/>
```

Correct

```
<xsd:element name="DeliveryPresentationOrder" type="xsd:positiveInteger" minOccurs="0"/>
```

LoadNoOfItems

Lars Henriksson, Biometria, 2016-03-03

LoadNoOfItems is today a positiveInteger. It was decided to change it to Integer since this makes it comparable with GreenMass and Volume.

LongTimberDefinition

Peter Sundlöf, Jonas Byström and Stefan Bohlin (2019-10-09).

The basic problem is that that in several administrative systems the default value for LongLogButtMAX is 0 mm.

It was decided that the datatype of LongLogButtMAX is to be changed to positiveInteger.

Data type for TrackingKey

Heikki Komulainen, Ponsse (2019-08-20).

Data type is not set in TrackingKey element under complextype TrackCoordinatesType. The data type must be xsd:positiveInteger as all other key elements.

It was decided that the datatype of TrackingKey is changed to positiveInteger.

Data type for BarkParameters

Halvard Olderlund, Komatsu (2020-02-14).

It was decided suggested that ConstantA and FactorB (bark elements) both are changed to decimal from nonNegativeDecimal.

FORMAT OF USERIDS

Heikki Komulainen, Ponsse (2019-08-20)

Ponsse have had several problems with UserID format over the last few years. The SF2010 implementation is database based and some of these issues are associated with reserved characters of database, some of these are our bugs, and

there is also a few times when incoming message contains “invalid” userID. Some examples:

- Whitespace characters before or after the actual userID value
 - Our systems removes the whitespaces
- UserID value contains only whitespaces (one or more)
 - → Our system removes whitespaces, so result is empty string
 - In UI this value looks empty, even we do not remove the whitespace
- UserID contains database reserved characters like percentage or underscore.
 - These characters affects problems when doing database level comparison of two userIDs
- Character casing (uppercase/lowercase) issues.
 - This was our own bug, SF2010 already defines that userid comparison is case intensive.

Of course, all of these issues can be handled in a way or another in code, but some other manufacturer may do it in a different way so in long run it is much better if SF2010 schemas contains some more accurate constraints.

Ponsse did not have any proposal for new constraints, but at least length of the userid and allowed/prohibited characters would be good to define.

Comment Juha-Antti Sorsa, 2020-09-04

The following simple type (created by Juha-Antti Sorsa) will forbid the characters: % ? < > : " ' / \ | * + and only whitespace character that is allowed is one space between valid substrings. Two more characters (' and +) added. No length restrictions are included.

It was suggested that the following type is used for all UserIDs:

```
<xs:simpleType name="RestrictedStringType">
  <xs:restriction base="xs:string">
    <xs:pattern value="([\S-[%&lt;&gt;:&quot;&apos;/\|*+]]+
      ( ([\S-[%&lt;&gt;:&quot;&apos;/\|*+]]+)*"/>
    </xs:restriction>
  </xs:simpleType>
```

There was a discussion regarding whether this is break from basic principles of xml according to Henrik Persson. However, both Niklas Ericsson and Heikki Kumolainen pointed out that this causes a lot of problems and that it would improve the logical functionality of the standard.

Decided to discuss this issue at the next meeting.

LogLength

Mikael Eliasson, LogMax 2016-04-07

LogMax has noted when studying length measurements that it would quite often be an advantage to be able to adjust the calibration somewhere between 0 and +1 cm or between 0 and -1 cm. Since all length measurements are registered as an integer with unit cm (without any decimals) it is difficult to do a "perfect" adjustment.

It was suggested to update the definition LogLength in LogMeasurementHarvestingQualityControlType and LogMeasurementType to allow for decimals (nonNegativeDecimal). This means that log lengths in hpr and hqc are registered in the same way.

Henrik commented that we should limit the number of decimals.

Heikki raised the question of whether having a higher precision would be relevant due to the limitations in measuring techniques today.

Decided to discuss this issue at the next meeting.

STEMKEYS

Per Knutsson, CGI, 2020-09-08

Today the StemKey has data type xsd:PositiveInteger. There is no limitation on the size of this datatype. However, databases and development tools normally has a limitation on the size of integers.

In some cases the StemKeys implemented in some controllers has very large values. In order to avoid having uncertainty about whether a specific data type is large enough it is suggested that Keys are given a maximum value, e.g. comparable with int64 or bigint (up to 9223372036854775807).

Decided to discuss this issue at the next meeting in conjunction with the idea of implementing new simple data types as suggested by Henrik Persson.

SPELLING ERRORS

ReportFilterCategory

David Zackrisson, LogMax 2014-04-22

There is a "space" in enumeration "Object filter" (StanForD2010-CommonDefinitions_V3p0.xsd).

It was decided that the extra space is to be excluded.

Bark functions

Anders Nylund 2014-04-08.

Exclude country specific names for different bark functions.

Older bark functions excluded.

Postpone, to be dealt with jointly with possible addition of a new bark function during the spring 2021.

TopLogFreeBuck

Johan Möller, 2016-04-05

Suggested to change element name TopLogFreeBuck to MINDiameterSearch since this is what it really means.

Postponed as this is primarily a “visual” issue that does not improve the functionality.

ShiftKey

Teemu Paldanius, Ponsse (2018-11-01).

Element ShiftKey is misspelled (ShifKey) in OperatorShiftDefintion.

It was decided that the name of ShifKey is changed to ShiftKey.

UPDATE ELEMENT DESCRIPTIONS

Description of message type for feller buncher

Jonas Thorsson, 2020-08-11

It was decided that the description of messageType in boi schema should be updated:

Enumeration list of file extensions of StanForD 2010 messages: Allowed value in Object Instruction message is: "boi"~~"fbi"~~

It was decided that the description of messageType in bpr schema should be updated:

Enumeration list of file extensions of StanForD 2010 messages: Allowed value in Object Instruction message is: "bpr"~~"fbr"~~

LengthClassAdjustment description

Räsänen Tapio, August 12th, 2020. Based on a question from Ari Ronkainen / UPM.

Is this a spelling error in the documentation text (pin, version 3.1 and all other versions too)? I suppose it is and should be corrected.

Present description of element LengthClassAdjustment is:

Allowed list values are

"LengthClassAdjustment1" = Log belongs to first length class smaller or equal to log length (427 cm => class 400 cm)

"LengthClassAdjustment1" = Log belongs to closest length class, normal rounding, (427 cm => class 430 cm)

It was decided that the second “LengthClassAdjustment1” is changed to “LengthClassAdjustment2” in the decription of element LengthClassAdjustment.

ELEMENTS AND ATTRIBUTES TO BE DELETED

Need for agency

Is attribute “agency” needed for “UserIDs? Discussed 2014-11-09.

It was decided that attribute “agency” is excluded since there is no relevant use for it.

ControlStemSettings in bq

John Arlinger, Skogforsk, 2020-05-11.

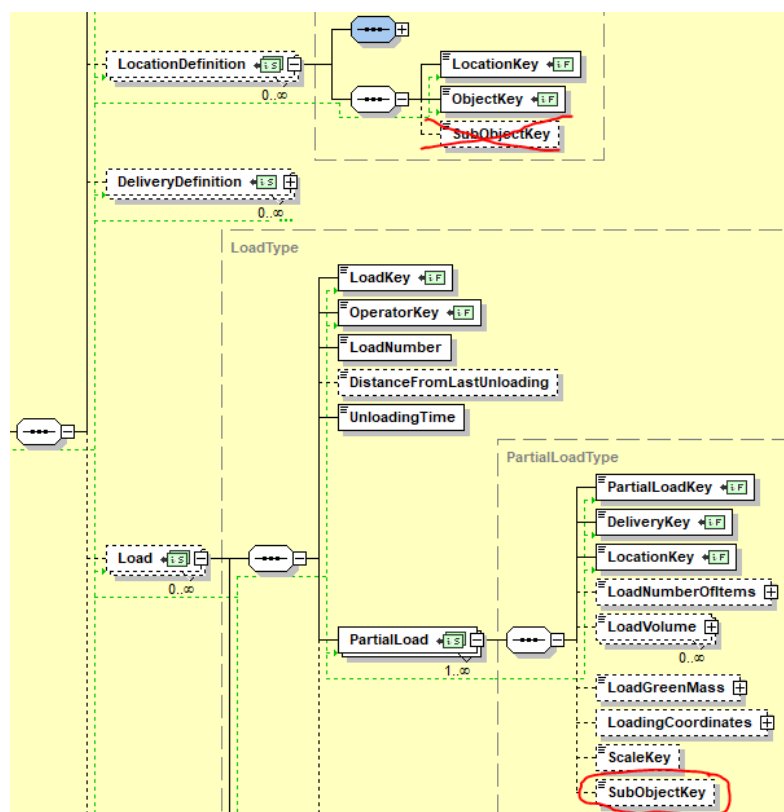
A large number of parameters in ControlStemSettings are irrelevant for feller bunchers since they depend on logs and/or length measuring.

It was decided that element RandomMINLength and RandomLogNotification are excluded from bq schema.

SUBOBJECT AND LOCATION

Fredrik Tjernberg, Komatsu, 2017-08-28

SubObjectKey is included under LocationDefintion. However, SubObjectKey was also added for registering under PartialLoad (2017-11-22) since having it under LocationDefinition makes it impossible to use one Location for several SubObjects. It was decided that SubObjectKey is deleted from LocationDefinition.



CARDINALITY (OPTIONAL VS MANDATORY)

LoggingOrganization

Teemu Paldanius, Ponsse, 2017-11-22

It was decided that ContactInformation is to be made optional in LoggingOrganisation with version 4.0

Country codes

Juha-Antti Sorsa, Metsäteho, 2017-10-25

Attribute countryCodeCategory was added as an optional attribute with version 3.4. The allowed enumeration value is "ISO 3166".

It was decided to make countryCodeCategory mandatory.

SubObject

Discussion StanForD-meeting 2017-11-22

SubObjects are mandatory in several machine systems. This means that a SubObject is always created per default when creating a new object.

It was suggested that to make SubObjectKey (under Stem) and SubObject (under Object) mandatory in files sent from forest machines in 4.0.

SubObjects are used by some Swedish companies, e.g. SubObjectUserID is registered by Biometria,

An idea raised at the meeting was to include a flag indicating if a SubObject has been automatically created by default.

Decided to discuss this issue at the next meeting.

CONSTRAINTS

FDI without ProductDefinition

Fredrik Tjernberg, Komatsu, 2015-11-04

It has been noted that it is possible to create an fdi-file where the DeliveryDefinition includes a ProductUserID but without including a ProductDefinition and SpeciesGroupDefinition as illustrated below:

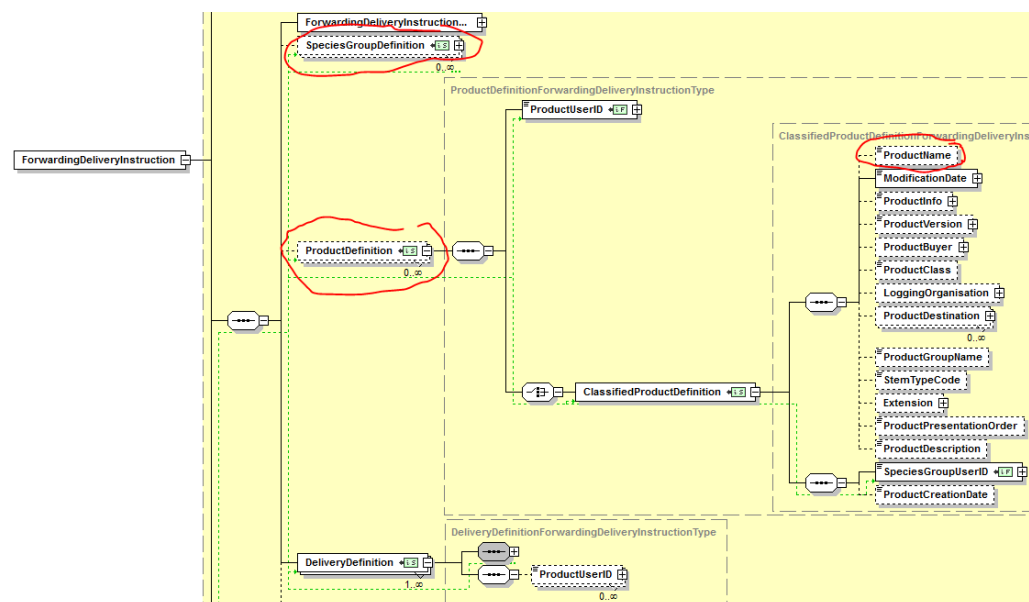
weightUnit	kg	
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance	
xsi:schemaLoca...	urn:skogforsk:stanford2010 ForwardingDeliveryInstruction_V3p0.xsd	
ForwardingDeliveryInstructionHeader		
SenderApplication	[REDACTED]	
CreationDate	2015-09-08T14:59:27+03:00	
ModificationDate	2015-09-14T14:28:45+03:00	
ApplicationVersionCreated	[REDACTED]	
ApplicationVersionModified	[REDACTED]	
CountryCode	[REDACTED]	
DeliveryDefinition (9)		
DeliveryUserID	ModificationDate	ProductUserID
1 SE_1604E_M2570_R003006_Y	2015-10-09T14:03:56+03:00	SE_1604E_M2570_R003006_Y
2 SE_1604E_T6700_R103007S_S	2015-10-09T14:03:56+03:00	SE_1604E_T6700_R103007S_S
3 SE_1604E_T3600_R002001_Y	2015-10-09T14:03:56+03:00	SE_1604E_T3600_R002001_Y

It was decided with version 4.0 that the schema is updated in order to make it mandatory to include ProductDefinition and SpeciesGroupDefinition in case ProductUserID is included with the DeliveryDefinition.

It was thus decided to add the following constraints as suggested by Juha-Antti Sorsa (2015-11-17):

```
<xsd:key name="ProductUserIDProductDefinition">
    <xsd:selector xpath="sfd:ProductDefinition"/>
    <xsd:field xpath="sfd:ProductUserID"/>
</xsd:key>
<xsd:keyref name="ProductUserIDRefProductDefinition"
refer="ProductUserIDProductDefinition">
    <xsd:selector xpath="sfd:DeliveryDefinition"/>
    <xsd:field xpath="sfd:ProductUserID"/>
</xsd:keyref>
```

It was also decided to make ProductName mandatory in order to make certain that there is some kind of relevant information connected to the products.



OTHER SCHEMA UPDATES

ShiftKey

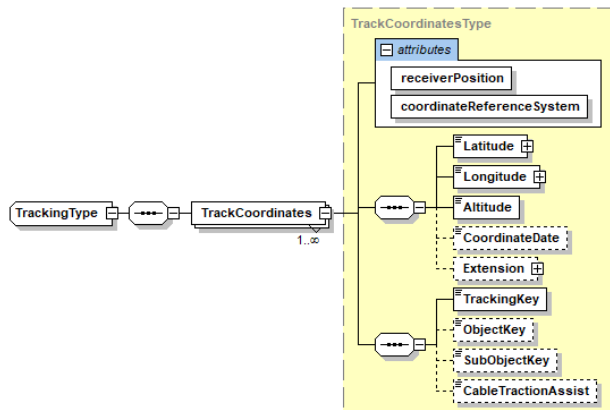
Issue discussed 2015-02-06

It was decided to add ShiftKey for each IndividualWorkTime as an optional element.

Tracking

Niclas Eriksson, Komatsu, 2017-11-29, Henrik Persson, LogMax, 2017-11-30 and 2019-02-28.

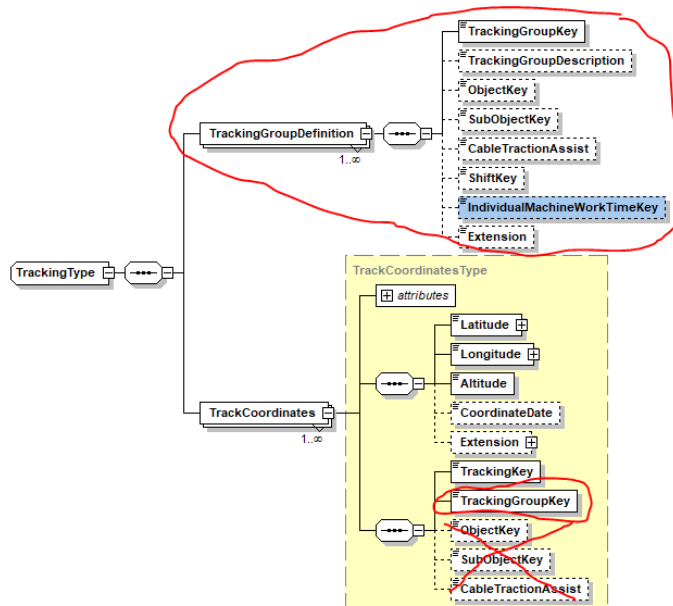
The present Tracking structure is illustrated below:



It was suggested that we study the possibility of introducing a new TrackingGroupDefinition with elements TrackingSettingKeys, ObjectKey, SubObjectKey, TrackingDescription, ShiftKey, IndividualMachineWorkTimeKey and Extension.

Observe that IndividualMachineWorkTimeKey does not exist today.

The figure below illustrates a possible new structure for registering additional tracking data:



Comments:

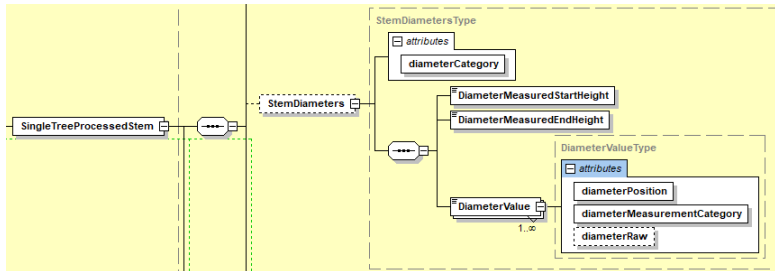
- The solution should be further developed to consider to keep frequently updated data such as IndividualMachineWorkTimeKey in the original TrackingCoordinates-structure.
- IndividualMachineWorkTimeKey does not exist today.

It was decided to discuss this issue at the next meeting

DiameterVector elements

Proposal Fredrik Tjernberg, Komatsu, 2015-09-07

Present structure for diameter vector in hpr illustrated below:



We could minimize the size of StemDiameter-vector by modifying element and attribute names according to the following list:

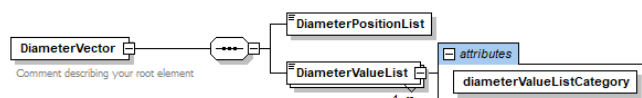
- DiameterValue = d
- diameterPosition = dp
- diameterMeasurementCategory = dc
- diameterRaw = dr

Old example files including 6140 stems:

5710085J-05_20120210_141903 Exkl DiaVektor.hpr	2015-09-03 11:19	HPR File	24 707 KB
5710085J-05_20120210_141903 Exkl DiaVektor.zip	2015-09-08 15:48	WinZip File	915 KB
5710085J-05_20120210_141903 Inkl DiaVektor.hpr	2015-09-03 11:34	HPR File	133 039 KB
5710085J-05_20120210_141903 Inkl DiaVektor.zip	2015-09-08 15:47	WinZip File	5 059 KB
5710085J-05_20120210_141903 Inkl minimeradDiaVektor.hpr	2015-09-03 11:38	HPR File	65 397 KB
5710085J-05_20120210_141903 Inkl minimeradDiaVektor.zip	2015-09-08 15:47	WinZip File	4 061 KB

Observe that the effect on a zipped file is significantly smaller than the effect on an uncompressed file.

Another solution (John Arlinger and Juha-Antti Sorsa, 2020-09-01) is to implement a list element as illustrated below. The individual values in the list are separated by a “space”. Observe that diameterValueListCategory includes the following enumeration values: Average, First, Second and Raw.



```
<xs:element name="DiameterVector">
  <xs:annotation>
    <xs:documentation>Comment describing your root element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="DiameterPositionList" type="IntValueListType"/>
      <xs:element name="DiameterValueList" maxOccurs="unbounded">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="IntValueListType">
              <xs:attribute name="diameterValueListCategory" use="required">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="Average"/>
                    <xs:enumeration value="First"/>
```

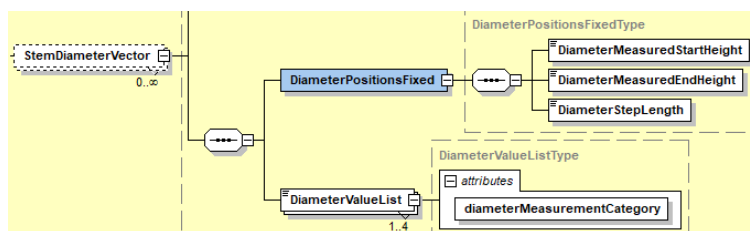
```

        <xs:enumeration value="Second"/>
        <xs:enumeration value="AverageRaw"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="IntValueListType">
  <xs:list itemType="xs:nonNegativeInteger"/>
</xs:simpleType>

```

DiameterVector	
xi:noNamespace SchemaLocation	TestSchema_DiameterVector_v2.xsd
xmlns:xi	http://www.w3.org/2001/XMLSchema-instance
DiameterPositionList	0 10 20 30 40 50 60 70
DiameterValueList	
diameterValueListCategory	Average
Text	100 99 96 96 95 89 75 74

Another even more compact solution could be to exclude the position list and replace the position list with a start and end height as well as a “step length” according to the following structure:



StemDiameterVector	
diameterCategory	Over bark
DiameterPositionsFixed	
DiameterMeasuredStartHeight	0
DiameterMeasuredEndHeight	70
DiameterStepLength	10
DiameterValueList	
diameterMeasurementCategory	Average
Text	100 99 96 96 95 89 75 74

It was suggested that we implement the last alternative and not both. The first alternative is more flexible and the second more compact.

Henrik Persson stated that using a list-construction means a step away from the basic xml-principles where, e.g. xltls (stylesheets) is difficult to use. He stated that he would prefer the solution suggested by Fredrik Tjernberg.

Jonas Thorsson and John Arlinger pointed out that the list structure decreases the file sizes and improves parsing speed.

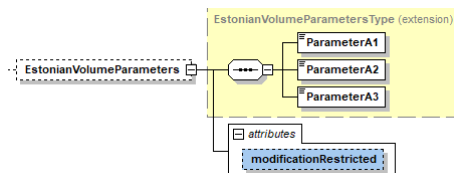
It was decided to discuss this issue at the next meeting and that John Arlinger will analyse the different alternatives in a bit more detail.

modificationRestricted

EstonianVolumeParameters

Stanford meeting spring 2016,

It was decided to add an attribute modificationRestricted for all EstonianVolumeParameters and a new element EstonianVolumeParameters:



ProductUserID

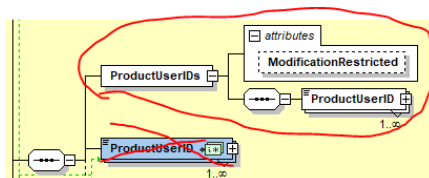
Jonas Gustafsson, Antti Korkiakoski, Erik Kindlund, Johan J Möller och John Arlinger (2017-09-27)

Patrik Anderchen (2020-03-12)

There is no attribute modificationRestricted affecting ProductUserIDs in an oin-file.

The idea is to add a flag to the oin that forbids changes to the list of products (ProdUserID). Some administrative systems require that products in oin and hpr are matching. In the new VIOL3 system developed by Biometria the harvester is only allowed to report products that has been specified for a specific contract. This means that if the operator adds an additional product the hpr will be rejected by the receiving system.

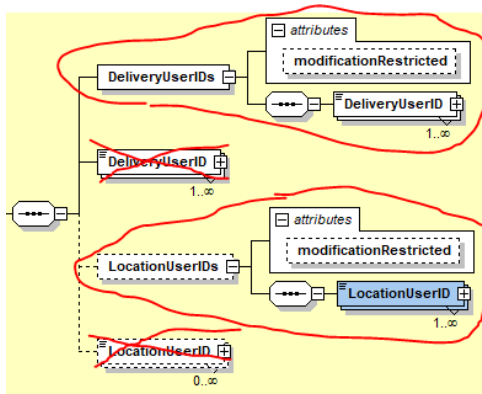
It was decided to add an attribute modificationRestricted for all ProductUserID and a new element ProductUserIDs:



DeliveryUserID and LocationUserID

There is no attribute modificationRestricted affecting DeliveryUserIDs and LocationUserIDs in an foi-file.

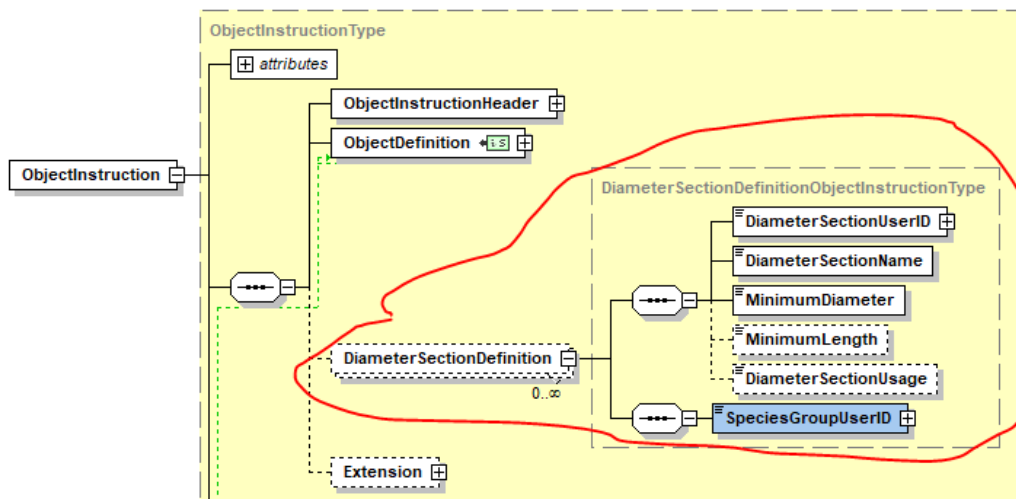
It was decided to add an attribute modificationRestricted for all DeliveryUserID and LocationUserID under new elements DeliveryUserIDs and LocationUserIDs:



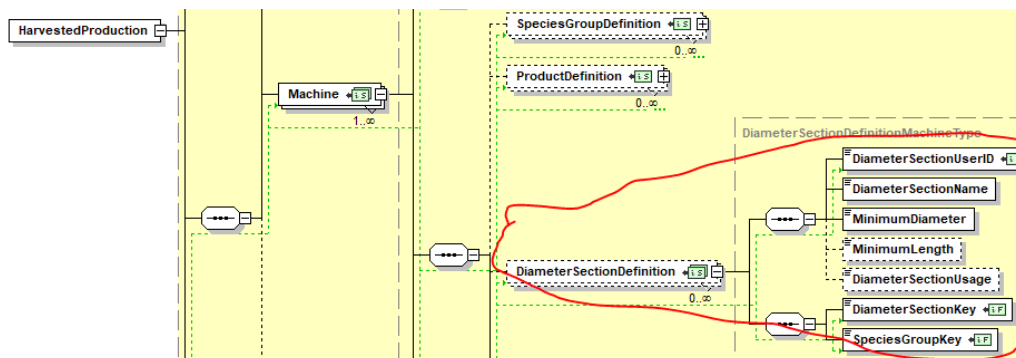
DiameterSectionUserID

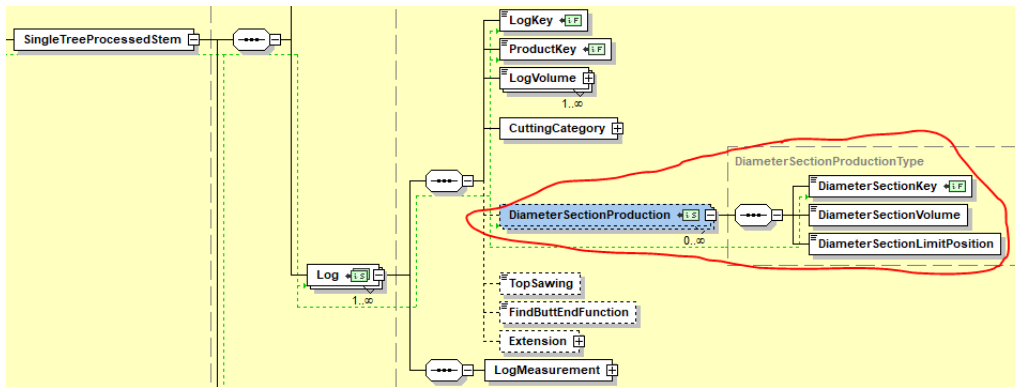
StanForD-meeting 2018-11-20 and Teemu Paldanius, Ponsse, 2018-11-01.

The definition of the DiameterSection is included in the oin sent to the harvester:

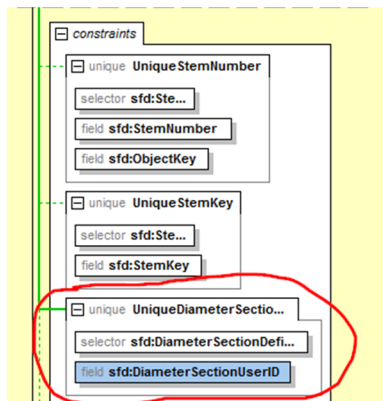


The following two figures illustrates how the DiameterSection is included in the hpr:





There is a constraint in the hpr message that DiameterSectionUserID must be unique under Machine element. It means that it is not possible to “update” DiameterSections by giving new DiameterSectionKey. It will lead to invalid hpr messages. Of course we could “replace” the old DiameterSectionDefinition and keep the Key, but we then lose the traceability if calculation parameters changes. And, because this constraint is under Machine level, we can’t update DiameterSections even between Objects.



It was decided to remove the DiameterSectionUserId constraint from the hpr and keep the unique DiameterSectionKey constraint.

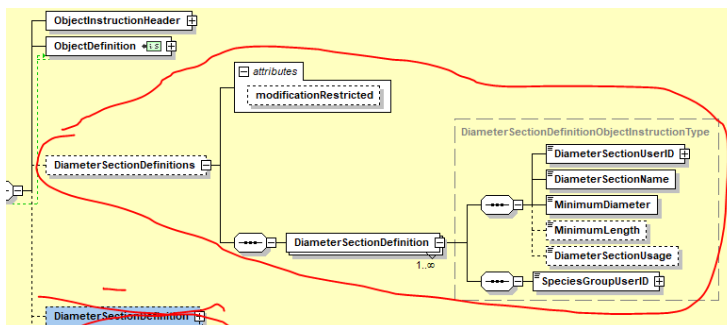
It was decided to move present DiameterSectionDefinition to ObjectDefinition in the oin and hpr based on a proposal from the Finnish StanForD meeting (not included in meeting agenda):

- to link the production measured by diameter sections in hpr (DiameterSectionProduction of logs) unambiguously with DiameterSectionKey to DiameterSectionDefinition
- needed especially in cases if DiameterSectionDefinition would be updated within an Object (e.g. MinimumDiameter or MinimumLength is changed)
- so far DiameterSections have been managed outside Object in the harvester system.

It was also decided to add the following new attributes and elements in order to follow the possible updates of DiameterSectionDefinitions or to prevent making any changes to them:

- DiameterSectionModificationDate
- attribute modificationAuthor
- attribute modificationRestricted

It was decided to add an attribute modificationRestricted for all elements DiameterSectionDefinition under new element DiameterSectionDefinitions. This is the same type of construction as for the previously added modificationRestricted for ProductUserID, LocationUserID and DeliveryUserID:



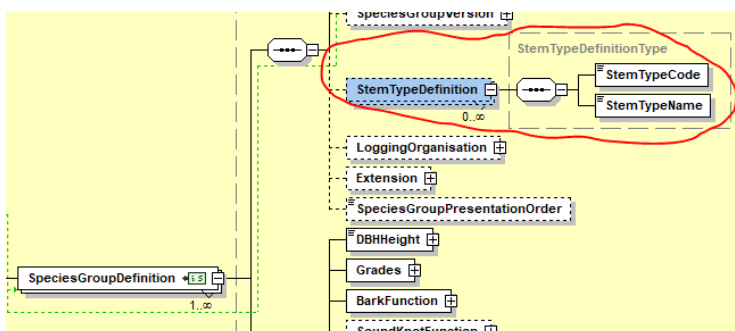
There was also a discussion regarding the needs for this type of calculated data since it could very well be calculated using the diameter vector. It was pointed out by the secretary that this type of calculated data would not be added to the standard today but that it is deemed difficult to remove it for historical reasons.

StemTypeCode

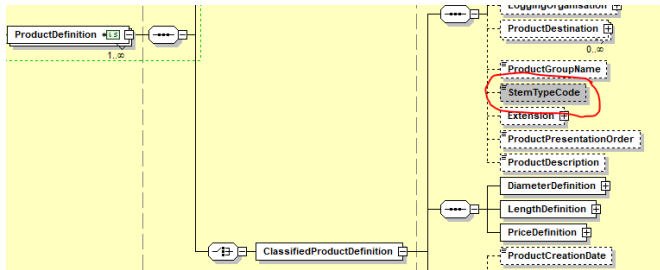
John Arlinger and Peter Sundlöf, Ponsse, 2019-02-01.

Structure of StemType probably should be changed since there is no referencing between spi and pin as in most other similar cases.

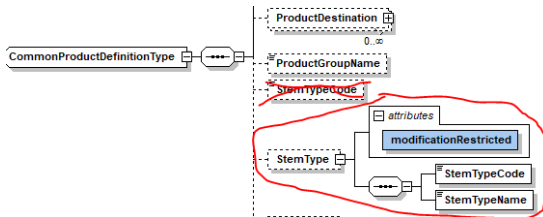
Present Spi:



Present Pin and Hpr:



It was suggested where StemTypeCode and StemTypeName is included in the pin and totally excluded from the StemTypeDefinition in the spi.



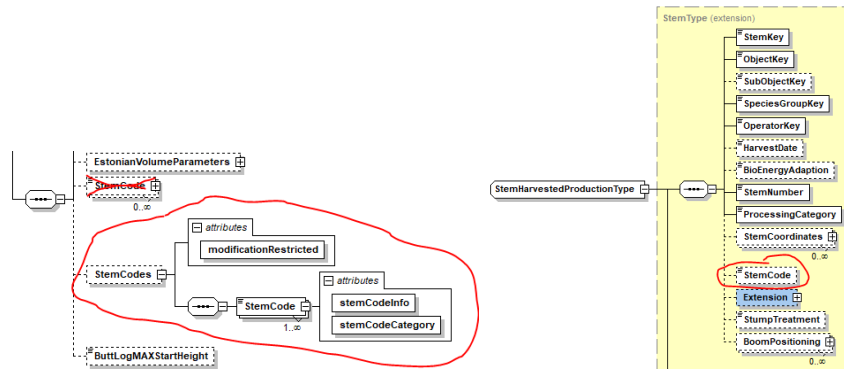
It was decided to discuss this issue at the next meeting in order to give Finnish actors the chance to study this issue in more detail

StemCode

John Arlinger and Johan Möller 2020-09-03. Niclas Eriksson, Komatsu, 2020-10-23.

Maybe add StemCodeInfo to be used for registering a code.

It was suggested to add the following structure to the SpeciesGroupDefinition as well as adding the StemCodeKey to element Stem:



Suggested standardized enumeration values in attribute stemCodeCategory:

- Nature (high stumps, retention trees etc)
- Culture (ruin, foot path etc)
- Special harvesting (selective cutting, windthrow, snow breakage etc)
- Other

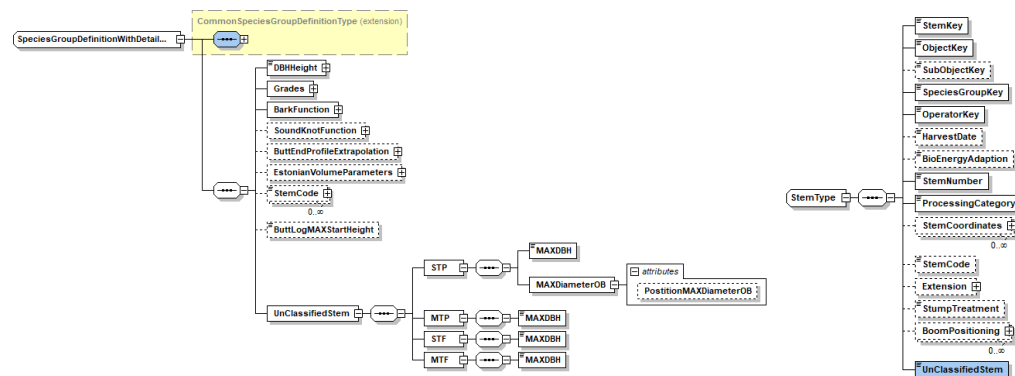
It was decided to discuss this issue at the next meeting, this new structure in the spi will be added to the 4.0 beta schema.

Unclassified stems

Anders Nylund, 2020-10-02.

Make it possible to define a flag for unclassified stems, DBH-limit per species, maybe also a diameter (pb) limit on specific height. If stem falls within these limits a flag "UnclassifiedStem" is set to true in hpr for the stem, otherwise false.

Mandatory elements in both Stem and SpeciesGroupDef. A more transparent solution comparing to having it as a machine setting. Makes it simple to exclude stems from different types of reports.



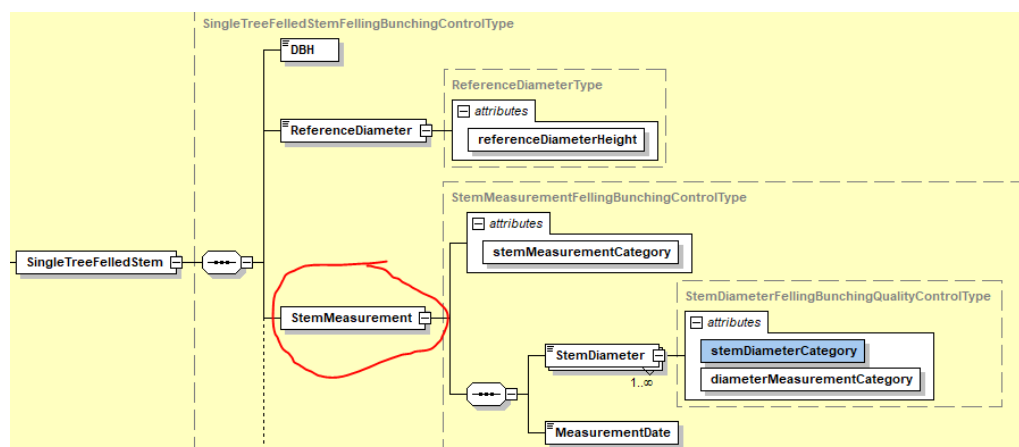
Discussed whether this type of “reporting parameters” should be included or not and whether we should create a separate message. Could include also other types of settings. Another comment was that **modificationRestricted** is missing in the proposal. There were also many participants pointing out that all trees/data MUST be stored in the hpr-file according to the standard. However, it is possible to present the trees in different groups in different types of reports.

It was decided to discuss this issue at the next meeting. Observe that this is not necessarily a 4.0 issue. Could be added also with 4.1 version.

Error in bqsc

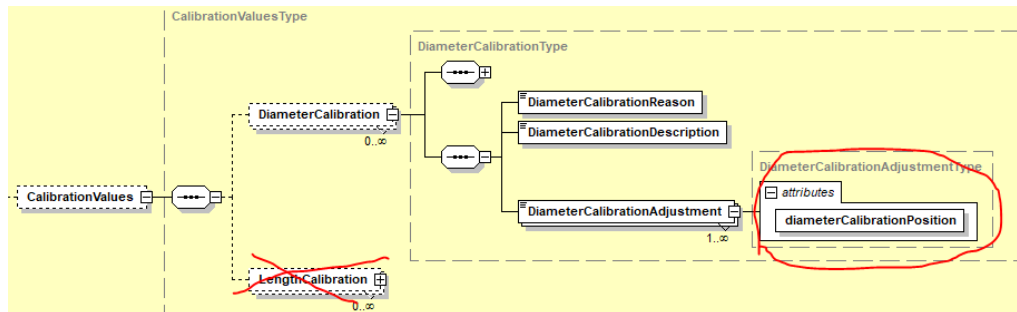
A serious error in the bqsc schema has been found after the meeting.

It was decided that element **StemMeasurement** is made repeatable. The bqsc schema is basically unusable as it is since it is impossible to include both machine and operator measurements.



Element CalibrationValues has structural problems.

It was decided that element LengthCalibration is excluded and that diameterCalibrationPosition is changed to stemDiameterCategory (values DBH or ReferenceDiameter)



SIGNIFICANT RE-STRUCTURING OF OGI-SCHEMA

Secretary gave a brief presentation of proposal (appendix X).

No objects or comments except for type of structure to be used for storing geometries. The main alternatives being GeoJson (as a string) or an XML-structure similar to KML.

It was decided to discuss this issue at the next meeting.

SIGNIFICANT RE-STRUCTURING OF MOM-SCHEMA

Secretary gave a brief presentation of proposal (appendix X).

New structure for registering down time reasons was discussed. Komatsu today have the possibility to set the down time reasons differently on different markets. The secretary will discuss this type of method with Komatsu.

It was decided to discuss this issue at the next meeting.

SIGNIFICANT RE-STRUCTURING OF FPR-SCHEMA

Not presented due to running out of time.

It was decided to discuss this issue at the next meeting.

IMPLEMENTATION RULES

Not discussed due to running out of time. Decided that it will be discussed at future meetings.

§7 Items for discussion (Appendix 4)

DISTRIBUTION BUCKING AND MULTI TREE PROCESSING

Not discussed due to running out of time. Decided that it will be discussed at future meetings.

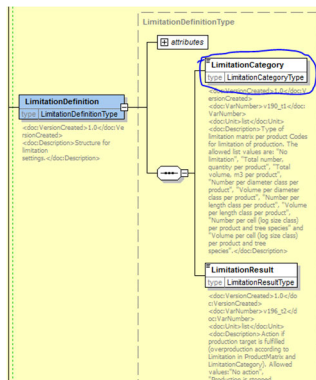
§8 Finnish proposal

NEW ENUMERATIONS TO LIMITATIONCATEGORYTYPE

Proposal raised by Metsä Group and Stora Enso and prepared by Trimble Forestry (Jyrki Vainio). Discussed at Finnish StanForD meeting 2020-10-29.

Proposal for new type of limitation matrix was discussed. This is done by adding new enumeration values to LimitationCategoryType:

- Relative volume in cell (log class) per total diameter class volume
- Relative number in cell (log class) per total diameter class numbers
- Relative volume in cell (log class) per total product volume
- Relative number in cell (log class) per total product numbers



Action in the bucking system

Intention with the new types is that the quantity of logs to be limited (number or volume of logs). The limitation is to increase as the total quantity of product (or diameter class) increases.

As the total quantity increases, logs that can be cut up to the set limitation percentage according to what has been set in LimitationResult

Example

- Dimensions of log class (cell)
 - length = 310 cm
 - diameter = 150 mm
- LimitationCategoryType = "Percentage of number of logs in cell per diameter class"
- Limitation = 5%

Number of cut logs in diameter class 150 mm	Maximum number of logs allowed to cut in cell (log class 310 cm & 150 mm) according to LimitationCategoryType and Limitation value
20	1
40	2
60	3

Issues to be considered

How does this kind of limitation work in the beginning of the Object? Especially if the quality of the stems is poor. A common technical solution should be specified for this.

Action in the bucking system when a new Object is started or pin is updated? The idea is that the limitation should not be reset when a new object is started with the same pin. How to manage the limitation and distribution matrices of the production over the Objects in two cases?

1. pin is not updated
2. pin is updated

The control system should take into account the previous production after the last update of pin without resetting the limitation matrices when a new object is started.

Machine operators need to understand well how the limitation works in different cases.

It was decided to discuss this issue at the next meeting. It was suggested that that a practical analysis is done on the effects based on “real” hpr-data.

NEW ENUMERATION "WINCHANCHORING"

Proposal from Heikki Komulainen, Ponsse and discussed in the Finnish StanForD group meeting 2020-10-29.

There is a market need to report the downtime caused by winch anchoring and releasing in mom with a specific code:

- to be used both for harvesting and forwarding
- harvesting at slopes is more common in many parts of the world and winch work is an essential part of the daily operations

It was suggested that value “Winch anchoring” is added to OtherMachineDownTimeCodeCategory.

Decided that the proposal is included in the work on a new down time structure for mom files as described earlier.

§9 Implementation issues

STANFORD 2010 NOMENCLATURE

No new language updates have been received.

TIME SCHEDULE FOR IMPLEMENTATION OF STANFORD2010

General StanForD tools

Manufacturer	Present software version released on the market	Last StanForD-version supported	Lower versions not supported
Silvia7 (CGI)	2019.4.0.1	3.5	<=2.0
ForesterOffice (Dasa)	ForesterOffice 1.7.1	3.1	
SenderXC (SDC)	1.12	3.6	<=2.0
DP II (Haglöf)	SKALMAN 7.18	3.6	<1.0

Harvesters

Specify latest minor version that has been or is planned to be implemented.

Manufacturer	Present software version released on the market	Last StanForD-version supported	Lower versions not supported
Opti (Ponsse)	Opti4G 4.761	3.1	2.0 or lower
Timbermatic (JohnDeere)	TimberMatic H 2.4	3.6	2.0
LogMate (LogMax)	2.0.31	3.5	< 2.0
MaxiXplorer (Komatsu)	3.22 & MaxiXT 1.3	3.6	Presently not 3.2
DasaForester	Forester H70 1.7.1	3.5	<2.0
Motomit (Parker)			
xLogger (Technion)	7.81	3.3	< 3.0
TOC 2 (LogSet)	3.343	3.0	< 3.0

Implemented elements	Opti	Timber-matic	Log-Mate	Maxi-Xplorer	Dasa-Forester	Motomit	xLogger	TOC 2
ProductDef.ProductClass	X	X	X		X		X	
Stem.StemCode			X	X	X			
Machine.Tracking (gps-log)			X	X	X			
SingleTreeProc.Reversing			X					
Stem.BoomeAngle		X		X	X(Rottne)			
ModificationRestricted		X	X	X	X		X	
UDI				X				

Forwarders

Specify latest minor version that has been or is planned to be implemented.

Manufacturer	Present software version released on the market	Last StanForD-version supported	Lower versions not supported
Opti (Ponsse)	Opti4G 4.761	3.1	2.0 or lower
Timbermatic (JohnDeere)	TimberMatic F 2.4	3.6	2.0
LogMate (LogMax)	No forwarder application		
MaxiXplorer (Komatsu)	3.22 & MaxiXT 1.3	3.6	Presently not 3.2
Forester (Dasa)	Forester F30 1.4	3.3	<2.0
Motomit (Parker)	No forwarder application		
xLogger (Technion)	No forwarder application		
TOC 2 (LogSet)	No forwarder application		

Implemented file types	Opti	Timber-matic	Maxi-Xplorer	Dasa-Forester	Motomit	xLogger	TOC 2
------------------------	------	--------------	--------------	---------------	---------	---------	-------

Forwarding instructions (foi and fdi)?	X	X	X	X			
--	---	---	---	---	--	--	--

§10 StanForD2010 projects / implementations

TEST WITH CALIPER CONTROL IN HQC-EXTENSION

Not discussed due to running out of time but information can be found in appendix 5.

POSSIBLE FORWARDER TEST IN SWEDEN (SVEASKOG AND SKOGFORSK)

Johan Möller presented an idea to arrange a forwarder test in Sweden during late spring 2021. The test will be carried out together with Sveaskog. An invitation will be sent out to all manufacturers before Christmas.

The idea is to run the test including the following parts:

1. New object using manual registration of unloaded volume
 - Start with complete Envelope include pre-defined locations.
 - Open and activate updated Delivery (fdi).
2. New object using manual registration of unloaded volume
 - New Delivery (fdi) added
 - Start with new foi without any correctly defined locations
 - Open and activate updated foi (additional Delivery).
 - Manually create locations with correct coordinates.
 - Take an updated foi (new delivery added) into use.
3. New object using registration of unloaded mass with scale and testing scale control (fqc) if scale measurements are available
 - Start with new foi using SenderXC
4. New object using automatic registration of unloaded volume based on hpr-data (if GIS-functionality is available)
 - Start with new foi
 - Control if volume is included in mom
5. Other tests
 - Control conversion between mass and volume.
 - Check if tracking is included
 - Evaluate GUI for delivery vs product
 - GIS-application (hpr-data presented independently of manufacturer brand)
 - Correct identities in fpr and mom.
 - How is the delivery library working?

- Registration of negative volumes possible?
- Simplicity when starting an object with a new foi.
- Other functionalities that the manufacturer wants to be tested/shown.

§11 Other issues

No other issues.

§12 Next meeting

2021-02-05, 10:00-15:00 CET, (only members) Teams

2021-04-21, 10:00-16:00 CET, Teams

Planning for a couple of additional Teams meetings during 2021.

By default, all financially contributing members will receive an Outlook invitation to all meetings in advance.

§13 End of meeting

The chairman thanked ever one for attending the meeting.

StanForD File REST-API version 0.1

NEW COMMUNICATION METHODS

DRAFT

Contents

Contents.....	1
Introduction	5
Base-line requirements on authentication.....	5
Authentication for local API.....	6
Authentication for remote API	6
Authorization.....	6
Request.....	6
Response data	7
StanForD data responses.....	7
List responses	7
Error responses.....	7
400 (Bad request).....	8
401 (Unauthorized).....	8
403 (Forbidden).....	8
404 (Not found)	8
405 (Method not allowed).....	8
Special cases with empty/no response	8
Compressing data in Response	8
API	9
/.....	9
Get.....	9
Post.....	9
Put	9
Delete	9
Head.....	9
/Capabilities	9
Get.....	9
Post.....	10
Put	10
Delete.....	10
Head.....	10
/File/v0.1.....	10
Get.....	10
Post.....	10
Put	10

Delete.....	10
Head.....	10
/File/v0.1/status/synchronization/<BaseMachineManufacturerID>.....	10
Get.....	10
Post.....	11
Put	11
Delete.....	11
Head.....	11
Retrieving data from machine	11
/File/v0.1/HPR	11
Get.....	11
Post.....	11
Put	12
Delete.....	12
Head.....	12
/File/v0.1/HPR/<HPR-file-ID>	12
Get.....	12
Post.....	12
Put	12
Delete.....	12
Head.....	12
/File/v0.1/HQC	12
Get.....	12
Post.....	13
Put	13
Delete.....	13
Head.....	13
/File/v0.1/HQC/<HQC-file-ID>	13
Get.....	13
Post.....	13
Put	13
Delete.....	13
Head.....	13
/File/v0.1/MOM	13
Get.....	13
Post.....	14
Put	14

Delete.....	14
Head.....	14
/File/v0.1/MOM/<MOM-file-ID>	14
Get.....	14
Post.....	14
Put	14
Delete.....	14
Head.....	14
Sending data to machine	14
/File/v0.1/SPI.....	14
Get.....	14
Post.....	15
Put	15
Delete.....	15
/File/v0.1/SPI/<BaseMachineManufacturerID>	15
Get.....	15
Post.....	15
Put	15
Delete.....	15
Head.....	15
/File/v0.1/PIN.....	15
Get.....	15
Post.....	16
Put	16
Delete.....	16
/File/v0.1/PIN/<BaseMachineManufacturerID>	16
Get.....	16
Post.....	16
Put	16
Delete.....	16
Head.....	16
/File/v0.1/OIN.....	16
Get.....	16
Post.....	17
Put	17
Delete.....	17
Head.....	17

/File/v0.1/OIN/< BaseMachineManufacturerID >	17
Get.....	17
Post.....	17
Put	17
Delete.....	17
Head.....	17
Summary of file APIs	17
Example APIs.....	17
Sending pin-file	18
Retrieving hpr-file.....	18
Example access control list (authorization registry).....	19
Local API.....	19
Remote API.....	19
Example method for retrieving machine data	20
Decision process.....	20

Introduction

The work on developing a new communication interface for StanForD was started with an open meeting in August 2019. The work continued inside a smaller work group with representatives from most manufacturers as well as Skogforsk, Biometria, Metsäteho and UPM.

The StanForD REST-API is a variant of a RESTful data communications interface. This means that the communication should be state-less and consistent. This means that the same command on the same data should generate the same response every time.

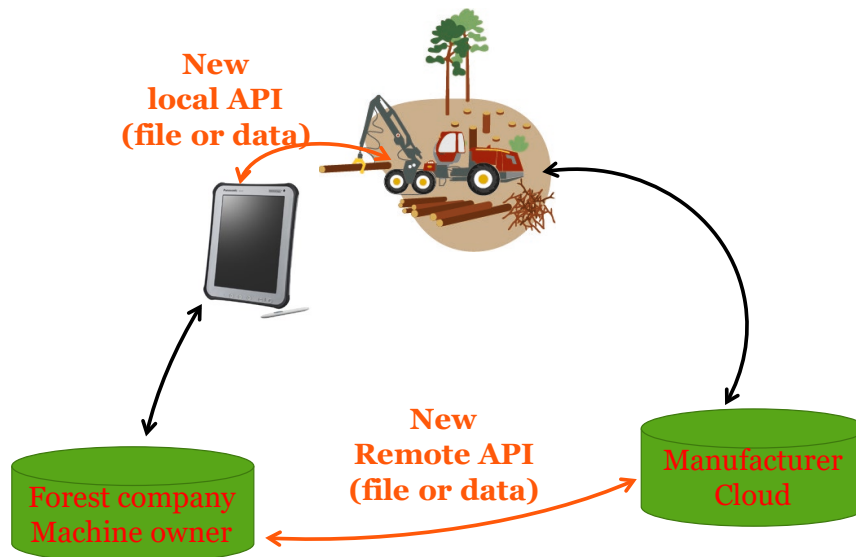
The API uses HTTPS as communication protocol. The HTTPS protocol handles the transmission security for the communication.

The minimum StanForD-version that can be used in this REST-API is 3.0. This means that e.g. versions 2.0 and 2.1 are not supported.

The following terms are used in this document:

- File API, an API for re-distributing a StanForD file conforming to the StanForD2010 schemas. The file is normally created by control system independently of API.
- Data API, an API for extracting a dataset based on the StanForD2010 standard but that may only include a subset of data comparing to the complete schemas.
- Local API, an API used for communication between device inside machine and control system of machine.
- Remote API, an API used for communication with a server/cloud outside machine.

The differences between local and remote APIs is illustrated below.



Base-line requirements on authentication

- TLS (HTTPS) is a requirement on all communication for remote services.
- Any connection to the API without proper authentication shall result in error code 401 (Unauthorized).
- The same API calls (requests) to be used in local and remote APIs. This means that the same logic is to apply in both cases.
- The access control list must define what types of files a user can retrieve.

- It must be possible to also have users (normally machine owner) that can access all LoggingOrganisation.BusinessIDs using some type of wild card.

Authentication for local API

The authentication for a local API is done through HTTP Basic Authentication (RFC7617) and is included in the HTTP header of every question. The Realm shall always be “StanForD”. The User-id field is optional, and not necessary in the authentication. If User-id is not used, the colon “:” before the Password shall be the first character in the sequence to be Base64 encoded. This will make it possible to embed an API-key in the HTTP header as a Password without a User-id.

A local API is assumed to always be used on local network behind some type of firewall.

Authentication for remote API

Different possible authentication solutions have been discussed for remote APIs. No decision has been possible to reach regarding a single standardized solution. It has thus been decided to establish the following baseline requirements for remote API authentication:

- Different manufacturers may implement different frameworks for authentication in remote services.
- Calls to remote services should not be restricted to one single domain. Small forest companies might be using several different service providers such as Biometria, CGI and temporary IT consultants.
- Basic authentication not recommended to be used in remote APIs.

Authorization

All data authorization is based on Logging Organization. Each client login is associated with an access control list for each machine, which authorize different clients to different Logging Organizations. The element ObjectDefinition/LoggingOrganisation/ContactInformation/BusinessID in each data type are then used to identify which Logging Organization that owns the data, and thereby which clients that shall have access to it.

Files without or with empty element LoggingOrganisation/ContactInformation/BusinessID are accessible only to users that have access to all logging organizations (e.g. a machine owner).

Files including several ObjectDefinitions with different ObjectUserIDs will be made available for all included logging organizations in the case of using file APIs.

An example of an access control list is included at the end of the document.

The access control list (registry) must define what types of files a user can retrieve.

Request

Requests to the server shall be as HTTP requests to the server URL using one of the following methods GET, POST, PUT, DELETE or HEAD. The URL and methods are described in the chapter API.

Each request must contain the client authentication, to keep the state-less nature of the protocol.

Parameters sent to the server in GET requests are sent in the URL of the API call.

Parameters sent to the server in POST requests are sent URL encoded in the http request header.

Data sent to the server is sent in the http request body in the form of StanForD2010 xml data.

Header must include:

- Content-Type “application/xml” or “text/xml”.
- Content-Length which indicates the size of the body.

- Content-Encoding if compressed files are included.
- Content-Disposition if communicating files as illustrated below:
`setHeader("Content-Disposition", "attachment; filename=\"filename.xml\"");`

All date parameters (StartTime and EndTime) are to be according to UTC as illustrated below:
 YYYY-MM-DDTHH:MM:SSZ e.g. 2020-10-01T15:22:09Z

Exactly the same requests to be used in local and remote APIs. This means that the same logic is to apply in both API types.

Response data

All requests to the server shall generate a response. The minimum response is an HTTP message with an Error code and reason-phrase in the status-line according to RFC7230 section 3.1.2. If the request generates response data, the data shall be sent in the message body.

Header must include:

- Content-Type “application/xml” or “text/xml”.
- Content-Length which indicates the size of the body.
- Content-Encoding if compressed files are included.
- Content-Disposition if communicating files as illustrated below:
`setHeader("Content-Disposition", "attachment; filename=\"filename.xml\"");`

All dates in responses are to be according to UTC as illustrated below:
 YYYY-MM-DDTHH:MM:SSZ e.g. 2020-10-01T15:22:09Z

There are three different response types (data, list and error) as described in sections below.

StanForD data responses

Data is sent in the message body in the form of StanFord2010 xml data.

List responses

When a request results in a reply from the server in the form of a list or collection, the response will contain an XML-formatted response message in the format:

```
<Response>
  <Entry>"Entry 1"</Entry>
  <Entry>"Entry 2"</Entry>
  ...
</Response>
```

Observe that the Entry list can be empty when e.g. no file or machine ids are available based on given parameters. The Entry list is also empty when posting files (spi, pin and oin) to a server.

Error responses

There are a number of error responses the server can give. Normal HTTP status codes are used. Every error response will contain an XML-formatted error message in the format:

```
<Error>
  <Code>"Error code"</Code>
  <Message>"Error message"</Message>
  <Entry>"Entry 1"</Entry>
  <Entry>"Entry 2"</Entry>
  ...
</Error>
```

The specific uses of the error responses are described in the API chapter.

400 (Bad request)

The error code 400 signals that the request contained syntactical or parameter range errors. The Message element shall contain a clear text description of the error reason. The faulty parameter(s) shall be listed in the Entry element(s).

401 (Unauthorized)

The error code 401 signals that the request lacked any authentication credentials. The Message element shall only contain the text “Unauthorized”.

403 (Forbidden)

The error code 403 signals that the file sent to the server in the wrong file format or file version. If the file format is faulty, the Message element shall contain the text “Unsupported file type”, and the Entry elements will contain the supported file types. If the file version is unsupported, the Message element shall contain the text “Unsupported file version” and the Entry elements shall contain the supported file versions.

404 (Not found)

The error code 404 signals that the resource that was requested via the URL does not exist on the server. The Message element shall only contain the text “Not found”. This error code is e.g. to be used if requested file does not exist.

405 (Method not allowed)

The error code 405 signals that the method in the request is not supported on that URL. The Message element shall contain the text “The <method> method is not supported on this resource” and the **Entry** element shall contain the valid methods for this URL.

Special cases with empty/no response

Response from server when posting a file:

- No response body to be returned at all. Status code 200 in header.

Response when no files or machines are available given certain parameters (e.g. within a time interval):

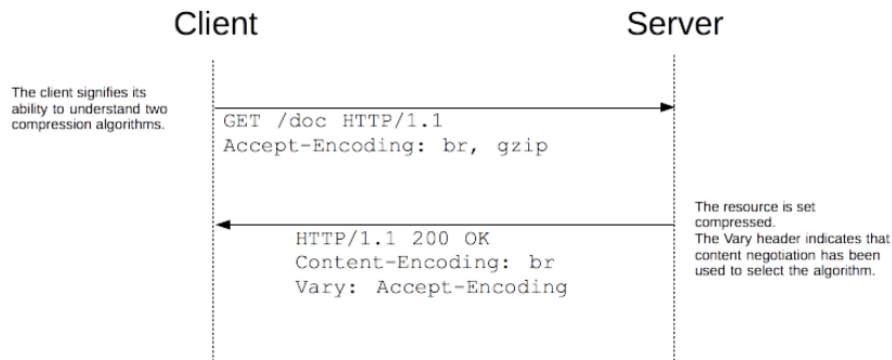
- <Response/>, i.e. no entry-elements included.

Response when a machine or file does not exist:

- Error 404

Compressing data in Response

Since StanForD-files can be of significant size it is relevant to be able to compress data included in the API response. There is functionality for compressing data built into http. The client can specify to the server what type of compression is accepted using http Accept-Encoding in the request header and the type of compression can be returned using Content-Encoding in the response header as illustrated below. The server shall by default compress data if accepted.



No other specific specification of compression methods is thus needed within this specification.

API

/

Root path of REST API. This node does not have to be the root node of the URL. I.e. the URL “https://www.example.com/StamForD/API/” might be the root node, as well as “https://api.example.com/”.

Get

Return a list of valid API versions. E.g. File_v0.1, File_v0.2. So far, only version 0.1 is available.

```

<Response>
  <Entry>"File_v0.1"</Entry>
  ...
</Response>
  
```

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

/Capabilities

Get

Returns a list of which API versions and StanForD file versions that are supported by the server. Also returns the number of days that a file is kept before deletion from server. The response is in the form of an XML data structure.

```

<Capabilities>
  <APIs>
    <API>"File_v0.1"</API>
    <API>"Data_v0.1"</API>
  </APIs>
  <FileVersions>
    <FileVersion>"3.0"</FileVersion>
    <FileVersion>"3.4"</FileVersion>
    <FileVersion>"3.5"</FileVersion>
  </FileVersions>
</Capabilities>
  
```

```
<FileDaysToExpiry>90</FileDaysToExpiry>
</Capabilities>
```

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

/File/v0.1

Root node of File API. Might be multiple versions available on same server. This makes it possible to develop the API and keep compatibility with older versions.

Get

Returns a list of file types that are supported by the server. E.g. HPR, MOM, SPI, PIN, OIN...

```
<Response>
  <Entry>"HPR"</Entry>
  <Entry>"MOM"</Entry>
  ...
</Response>
```

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

/File/v0.1/status/synchronization/<BaseMachineManufacturerID>**Get**

A request for the date and time of last complete synchronization between machine and remote server (no pending files in queue waiting to be sent to machine). The request is also useful when asking remote server for production files generated by the machine.

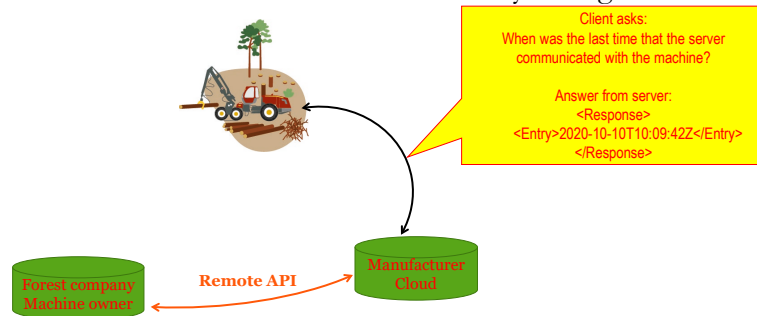
The request is only relevant for remote APIs and NOT for local APIs.

Returns a date and time for last synchronization.

```
<Response>
  <Entry>2020-10-10T10:09:42Z</Entry>
</Response>
```

If BaseMachineManufacturerID does not exist, the server returns Error 404 (Not found).

This GET method could be illustrated by the figure below:



Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

Retrieving data from machine

Hpr, hqc and mom are include below. The same type of calls are also to be implemented for other StanForD2010 files e.g. fpr, fq, bpr.

/File/v0.1/HPR

Get

A Get will return a list of available HPR-files (HPR-file-IDs). Available parameters to refine the list are; BaseMachineManufacturerID, ObjectUserID, StartDate, EndDate. If one or more parameters are given, the HPR-files with the selected search criteria are returned.

The default values for the parameters are; BaseMachineManufacturerID =<all>, ObjectUserId=<all>, StartDate=1970-01-01T00:00:00.OZ, EndDate=<now>.

The StartDate and EndDate parameters are looking at the CreationDate element in the HPR-files in local APIs while the time stamp when file is received by the server (ReceptionDate) is used in remote APIs. Only files which creation/reception dates are between the StartDate (CreationDate >= StartDate) and the EndDate (CreationDate< EndDate) will be included in the list.

```
<Response>
  <Entry>"98jk430dg2Q"</Entry>
  <Entry>"509xca97f7g"</Entry>
  ...
</Response>
```

No Entry elements are included in cases where e.g. no files are available given parameters used in the request:

```
<Response>
</Response>
```

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

/File/v0.1/HPR/<HPR-file-ID>**Get**

A Get will return a data stream containing the selected HPR-file data.

HPR-file-ID is never to change! Same for all other file-IDs.

Return Error 404 (Not found) if file-ID does not exist on the server.

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

A Head will return only the header which includes Content-Length. Can be used before GET in order to determine if file is too large to handle.

HPR-file-ID is never to change! Same for all other file-IDs.

Return Error 404 (Not found) if file-ID does not exist on the server.

/File/v0.1/HQC**Get**

A Get will return a list of available HQC-files (HQC-file-IDs). Available parameters to refine the list are; BaseMachineManufacturerID, ObjectUserID, StartDate, EndDate. If one or more parameters are given, the HPR-files with the selected search criteria are returned.

The default values for the parameters are; BaseMachineManufacturerID =<all>, ObjectUserId=<all>, StartDate=1970-01-01T00:00:00.0Z, EndDate=<now>.

The StartDate and EndDate parameters are looking at the CreationDate element in the HQC-files in local APIs while the time stamp when file is received by the server (ReceptionDate) is used in remote APIs. Only files which creation/reception dates are between the StartDate (CreationDate >= StartDate) and the EndDate (CreationDate< EndDate) will be included in the list.

```
<Response>
  <Entry>"925Q63gd43A"</Entry>
  <Entry>"45asdfv21eQW"</Entry>
  ...
</Response>
```

No Entry elements are included in cases where e.g. no files are available given parameters used in the request:

```
<Response>
</Response>
```

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

/File/v0.1/HQC/<HQC-file-ID>**Get**

A Get will return a data stream containing the selected HQC-file data.

HQC-file-ID is never to change! Same for all other file-IDs.

Return Error 404 (Not found) if file-ID does not exist on the server.

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

A Head will return only the header which includes Content-Length. Can be used to avoid downloading too large files when using the method GET.

HQC-file-ID is never to change! Same for all other file-IDs.

Return Error 404 (Not found) if file-ID does not exist on the server.

/File/v0.1/MOM**Get**

A Get will return a list of available MOM-files (MOM-file-IDs). Available parameters to refine the list are; BaseMachineManufacturerID, ObjectUserID, StartDate, EndDate. If one or more parameters are given, the MOM-files with the selected search criteria are returned.

The default values for the parameters are; BaseMachineManufacturerID =<all>, ObjectUserId=<all>, StartDate=1970-01-01T00:00:00.0Z, EndDate=<now>.

The StartDate and EndDate parameters are looking at the CreationDate element in the HQC-files in local APIs while the time stamp when file is received by the server (ReceptionDate) is used in remote APIs. Only files which creation/reception dates are between the StartDate (CreationDate >= StartDate) and the EndDate (CreationDate < EndDate) will be included in the list.

```
<Response>
  <Entry>"8ds32twlVd2"</Entry>
  <Entry>"G59xca97fV7g"</Entry>
  ...
</Response>
```

No Entry elements are included in cases where e.g. no files are available given parameters used in the request:

```
<Response>
</Response>
```

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

/File/v0.1/MOM/<MOM-file-ID>**Get**

A Get will return a data stream containing the selected MOM-file data.

MOM-file-ID is never to change! Same for all other file-IDs.

Return Error 404 (Not found) if file-ID does not exist on the server.

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

A Head will return only the header which includes Content-Length. Can be used before GET in order to determine if file is too large to handle.

MOM-file-ID is never to change! Same for all other file-IDs.

Return Error 404 (Not found) if file-ID does not exist on the server.

Sending data to machine

Oin, pin and spi are include below. The same type of calls are also to be implemented for other StanForD2010 files e.g. foi, fdi, boi and env.

Observe that the POST API for distributing a file to all machines within a LoggingOrganization is only to be implemented for spi, pin, fdi and not for oin, foi, boi or env.

/File/v0.1/SPI**Get**

A Get with no additional parameters will return a list of available BaseMachineManufacturerID. Since each machine is allowed to have different definitions of each Species, SPI-files must be handled on Machine level.

```
<Response>
  <Entry>"r6434565"</Entry>
  ...
</Response>
```

No Entry elements are included in cases where no machines are available:

```
<Response>
</Response>
```

Post

A Post with an optional BusinessID parameter set to a Logging Organization will add the SPI-file to the inbox of all Machines associated with the Logging Organization. A Post without BusinessID will add the PIN-file to all Machines that the user has access to. The Post must contain a valid SPI-file in the payload and can include optional parameter BusinessID.

No response required in case of successful Post (status code 200).

If the server cannot handle the StanForD-version of SPI-file, the server returns Error 403 (Forbidden).

If the user does not have access to the BusinessID, the server returns Error 400 (Bad request).

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

/File/v0.1/SPI/<BaseMachineManufacturerID>**Get**

Returns Error 405 (Method Not Allowed).

Post

A Post must contain a valid SPI-file in the payload. The SPI-file will be added to the Machines inbox.

No response required in case of successful Post (status code 200).

If the server cannot handle the StanForD-version of SPI-file, the server returns Error 403 (Forbidden).

If BaseMachineManufacturerID does not exist, the server returns Error 404 (Not found).

Put

Returns Error 405 (Method Not Allowed).

Delete

Returns Error 405 (Method Not Allowed).

Head

Returns Error 405 (Method Not Allowed)

/File/v0.1/PIN**Get**

A Get with no additional parameters will return a list of available BaseMachineManufacturerID. Since each machine is allowed to have different definitions of each Product, PIN-files must be handled on Machine level.

```
<Response>
  <Entry>"r6434565"</Entry>
  ...
</Response>
```

No Entry elements are included in cases where no machines are available:

```
<Response>
</Response>
```

Post

A Post with an optional BusinessID parameter set to a Logging Organization will add the PIN-file to the inbox of all Machines associated with the Logging Organization. A Post without BusinessID will add the PIN-file to all Machines that the user has access to. The Post must contain a valid PIN-file in the payload and can include optional parameter BusinessID.

No response required in case of successful Post (status code 200).

If the server cannot handle the StanForD-version of PIN-file, the server returns Error 403 (Forbidden).

If the user does not have access to the BusinessID, the server returns Error 400 (Bad request).

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

/File/v0.1/PIN/<BaseMachineManufacturerID>**Get**

Returns Error 405 (Method Not Allowed)

Post

A Post must contain a valid PIN-file in the payload. The PIN-file will be added to the Machines inbox.

No response required in case of successful Post (status code 200).

If the server cannot handle the StanForD-version of PIN-file, the server returns Error 403 (Forbidden).

If BaseMachineManufacturerID does not exist, the server returns Error 404 (Not found).

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

/File/v0.1/OIN**Get**

A Get with no additional parameters will return a list of available BaseMachineManufacturerID. Since each machine is allowed to have different definitions of each Object, OIN-files must be handled on Machine level. A specific OIN is thus never sent to all machines.

```
<Response>
  <Entry>"r6434565"</Entry>
  ...
</Response>
```

No Entry elements are included in cases where no machines are available:

```
<Response>
</Response>
```

Post

Returns Error 405 (Method Not Allowed)

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

/File/v0.1/OIN/< BaseMachineManufacturerID >**Get**

Returns Error 405 (Method Not Allowed)

Post

A Post must contain a valid OIN-file in the payload. The OIN-file will be added to the Machines inbox.

No response required in case of successful Post (status code 200).

If the server cannot handle the StanForD-version of PIN-file, the server returns Error 403 (Forbidden).

If BaseMachineManufacturerID does not exist, the server returns Error 404 (Not found).

Put

Returns Error 405 (Method Not Allowed)

Delete

Returns Error 405 (Method Not Allowed)

Head

Returns Error 405 (Method Not Allowed)

Summary of file APIs

Table below gives an overview of available APIs for different file categories. XXX stands for any of the specified file types.

File types	File/v0.1/XXX GET	File/v0.1/XXX POST	File/v0.1/XXX/ <XXX-file-ID> GET	File/v0.1/XXX/ <BaseMachineManufacturerID> POST
Hpr, hqc, fpr, fqc, bpr, mom	List of all available files	*	Specific file downloaded	*
Pin, spi, fdi	List of all available machines	File uploaded to all machines in LoggingOrganization	*	File uploaded to specific machine
Oin, foi, boi, env	List of all available machines	*	*	File uploaded to specific machine

*Method not allowed

Example APIs

Below follows an example communication between a client and a server where the client will send a PIN-file to all machines in logging organization 'BigCompany1' and then get the HPR-files generated between 2019-11-23 00:00:00Z and 2019-11-23 23:59:59Z.

Sending pin-file**Client to Server*****POST*** /File_v0.1/PIN?BusinessID=BigCompany1 ***HTTP/1.1****Host: fleet.example.com:443**Authorization: Basic VXNlcjE6MTIzNDU2**Content-Length: 29587**Content-Type: application/xml**Content-Disposition: attachment; filename="BigSamMill.pin"**<?xml version="1.0" encoding="utf-8"?><ProductInstruction...***Response from Server*****HTTP/1.1 200 OK****Content-Length: 0***Retrieving hpr-file****Client to Server*****GET*** /File_v0.1/HPR?StartDate=2019-11-23T00:00:00Z&EndDate=2019-11-23T23:59:59Z***HTTP/1.1****Host: fleet.example.com:443**Authorization: Basic VXNlcjE6MTIzNDU2**Content-Length: 0***Response from Server*****HTTP/1.1 200 OK****Content-Length: 91**Content-Type: application/xml**<Response>**<Entry>"Object1-1.hpr"</Entry>**<Entry>"Object1-2.hpr"</Entry>**</Response>***Client to Server*****GET*** /File_v0.1/HPR/Object1-1.hpr ***HTTP/1.1****Host: fleet.example.com:443**Authorization: Basic VXNlcjE6MTIzNDU2**Content-Length: 0**Accept-Encoding: gzip, br, deflate***Response from Server*****HTTP/1.1 200 OK****Content-Length: 3225343**Content-Encoding: gzip**Content-Type: application/xml**Content-Disposition: attachment; filename="Object1-1.hpr"**<?xml version="1.0" encoding="UTF-8"?><HarvestedProduction...***Client to Server*****GET*** /File_v0.1/HPR/Object1-2.hpr ***HTTP/1.1***

Host: fleet.example.com:443
 Authorization: Basic VXNlcyE6MTIzNDU2
 Content-Length: 0
 Accept-Encoding: gzip, br, deflate

Response from Server

HTTP/1.1 200 OK

Content-Length: 1832664

Content-Encoding: gzip

Content-Type: application/xml

Content-Disposition: attachment; filename="Object1-2.hpr"

<?xml version="1.0" encoding="UTF-8"?><HarvestedProduction...

Example access control list (authorization registry)

In order to keep track of the rights for accessing data some type of list must be kept by the server. The following example is split into two separate sections for local and remote APIs since there are some fundamental differences. The local API is only used for communicating with one single machine while the remote API can be used for communicating with a large number of machines simultaneously.

It is suggested that we recommend that these two access lists are kept separate (not synchronized) in order to avoid that changes to the same user can be made both remotely and locally. Having the possibility to view also the local list also by a remote solution is probably not a problem but the local access control list is to be edited in the machine.

Observe that additional limitations may very well be implemented such as a specific time interval, specific harvesting objects etc.

Local API

Access control list is set by operator in machine. User-ID and Password is probably presented in interface of control system and manually copied to local client.

Below are examples of access control lists for two separate machines.

BaseMachine-ManufacturerID: 5fd4nx643t56

User	Pass-word	BaseMachine-ManufacturerID	LoggingOrg.-BusinessID	File types
MachineOwner	8me02	5fd4nx643t56	* (All)	* (All)
Holmen	432g5	5fd4nx643t56	105	hpr, hqc, pin, spi, oin
Sveaskog	3j48cxv	5fd4nx643t56	106	hpr, hqc, pin, spi, oin
Södra	76jer6	5fd4nx643t56	110	hpr, hqc, pin, spi, oin
Biometria	87cso9	5fd4nx643t56	105	hpr, hqc, pin, spi
			106	hpr, hqc, mom, pin, spi
			110	hpr, hqc, pin, spi

BaseMachine-ManufacturerID: SS39jkfd435

User	Pass-word	BaseMachine-ManufacturerID	LoggingOrg.-BusinessID	File types
MachineOwner	7u65	SS39jkfd435	* (All)	* (All)
Sveaskog	Hert6	SS39jkfd435	106	hpr, hqc, pin, spi, oin
Biometria	5ejhd	SS39jkfd435	106	hpr, hqc, pin, spi
Storaenso	!5nn6	SS39jkfd435	349erds9	Mom, hpr, hqc

Remote API

Access control list set by machine owner.

User	Pass-word	BaseMachine- ManufacturerID	LoggingOrg.- BusinessID	File types
MachineOwner	63gt65	5fd4nx643t56	* (All)	* (All)
		SS39jkfd435	* (All)	* (All)
Holmen	Xy4h7	5fd4nx643t56	105	hpr, hqc, pin, spi, oin
Sveaskog	Xert6	5fd4nx643t56	106	hpr, hqc, pin, spi, oin
		SS39jkfd435	106	hpr, hqc, pin, spi, oin
Södra	X6jer6	5fd4nx643t56	110	hpr, hqc, pin, spi, oin
Biometria	Xejhd	5fd4nx643t56	105	hpr, hqc, pin, spi
			106	hpr, hqc, mom, pin, spi
			110	hpr, hqc, pin, spi
		SS39jkfd435	106	hpr, hqc, pin, spi
Storaenso	!Xnn6	SS39jkfd435	349erds9	Mom, hpr, hqc

It is possible, based on the access list above, to compile a list which defines which machines “belongs” to a specific logging organization. This type of list is relevant in cases when there is a need to upload e.g. a pin file to all machines working for a specific logging organization (POST /File_v0.1/PIN?BusinessID=106).

LoggingOrganization - BusinessID	BaseMachine - ManufacturerID
105	5fd4nx643t56
106	5fd4nx643t56 / SS39jkfd435
110	5fd4nx643t56
349erds9	SS39jkfd435

Example method for retrieving machine data

Observe that the only completely safe method to evaluate whether all data has been reported to the client is by checking that no StemNumbers or LoadNumbers are missing for a specific harvesting object.

A simple algorithm run every hour for retrieving and checking files could be to:

1. Check for all file-ids generated the last 24 hours.
2. Download all file-ids that has not been previously downloaded.
3. Import data into database
4. Check that no StemNumbers/LoadNumbers are missing from objects (a complete interval should exist from 1 to the last StemNumber in the file with EndDate).

Observe that the algorithm above is based on having a list with all historically downloaded file-ids.

Decision process

Decisions to release new version of the StanForD APIs will be made parallel with new StanForD2010 versions. This means that the first “release” version of this document will at the earliest be published in February 2021.

The technical work group will discuss how to start working on a data API.



Arlinger John
Ian Wilson

30 November 2020

New yarder and skidder messages

Introduction

New schemas for feller bunchers were added to the standard in 2018 (boi, bpr) and 2019 (bqc). Including other tree length machines, such as skidders and yarders, were discussed already in 2018 but it was decided to postpone these machine types.

It was suggested by Jeremy Gibson and Ian Wilson (2019-07-01) to start working on new schemas also for yarders and skidders.

The old production proposal (based on the fpr) for skidders from 2018 is illustrated below.

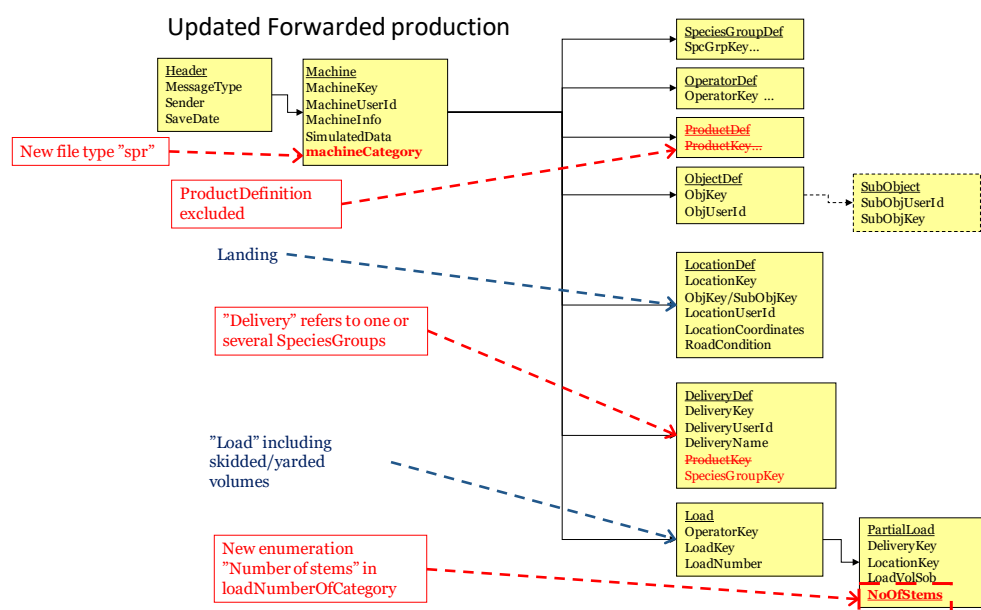


Figure 1. Early preliminary proposal for new skidder and yarder production message.

In discussions between Ian, Jeremy and John Arlinger it was deemed best to keep yarders and skidders separate as was also discussed at the meeting 2018-11-20 (table below).

Table 1. File extensions also to be used for tree length machines are described below.

	Object Instruction	Production Report	Quality Control	Operational Monitoring
Harvester	Oin+Pin+Spi	Hpr	Hqc	Mom
Forwarder	Foi+Fdi	Fpr	Fqc	Mom
Feller Buncher	Boi+Spi	Bpr	Bqc	Mom
Skidder	Soi+Spi	Spr	-	Mom
Yarder	Yoi+Spi	Ypr	-	Mom

This document includes proposals for production reports and object instructions. Adding support for these machines also in operational monitoring reports (mom) will be postponed until version 4.0

Skidder production

The basic idea is to report each skidder-load with information about species and number of stems. This means that the Deliveries and Products are irrelevant when comparing with the forwarder.

The following existing elements and attributes need new enumeration values:

- Add “spr” to attribute messageCategory
- Add “Skidder” to attribute machineCategory
- Add “Number of stems” to attribute loadNumberOfCategory

When comparing with the present standard for forwarder production the following new parameters are deemed to be required:

- **Winch**
 - Brand/Model/Year
- **Extraction Method**
 - Grapple
 - Choker/Winch
- **Stem Presentation**
 - Un-bunched – stems are generally not grouped into a bunch and are typically felled by a person with a chainsaw
 - Part Bunched – stems are roughly grouped into a bunch and are typically mechanically felled using a Dangle Head feller buncher
 - Bunched – stems are precisely grouped into a bunch and are typically mechanically felled using a Fixed Head feller buncher
 - Machine Fed – stems are placed or lifted by a separate machine into the grapple or for the grapple to grab them

It is suggested to add Winch information to the Machine and to add ExtractionMethod and StemPresentation to the Load as illustrated below. In most cases ExtractionMethod and StemPresentation are not changed within an object but they sometimes are depending e.g. on the terrain variability. This information is set by the operator.

Observe that the figure below is a simplification of the schema. The idea is to keep all elements and attributes from the fpr-schema except for those explicitly mention in previous section.

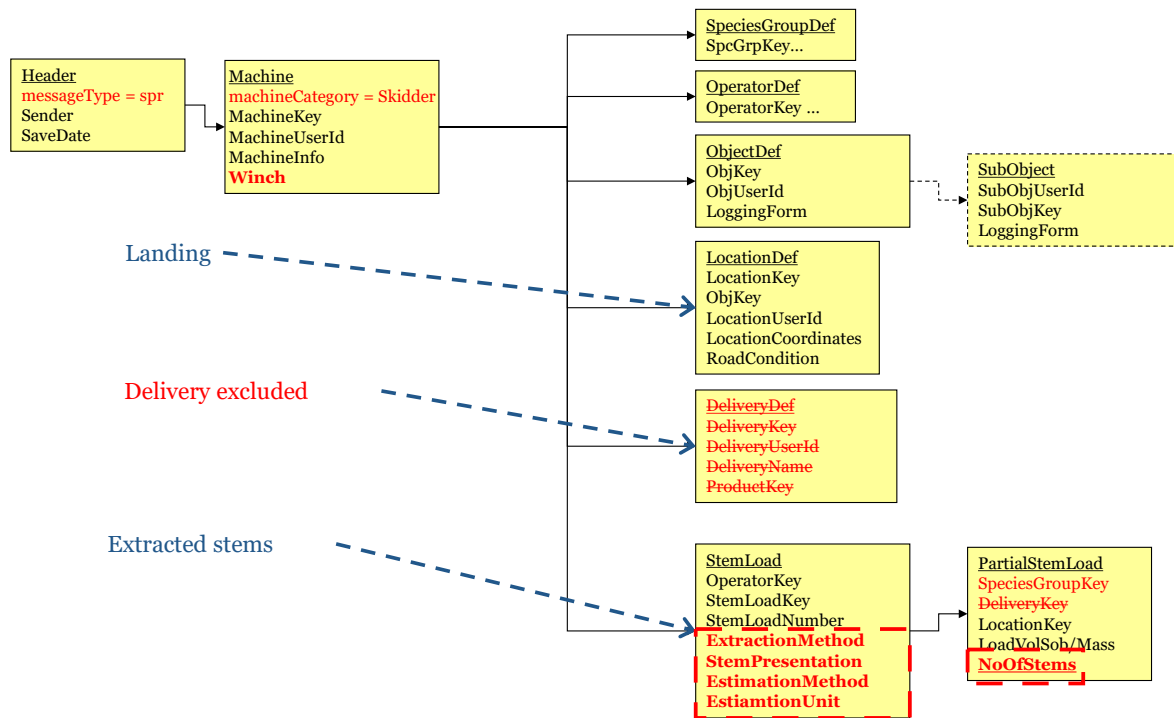
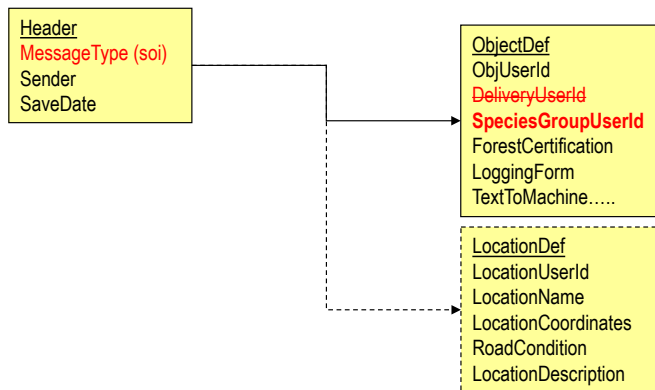


Figure 2. New or updated elements or attribute in red color. Observe that DeliveryDefinition is to be excluded.

Observe that machineCategory will not include the enumeration value "Skidder" until major version 4.0. Instead the value "Other" has to be used in the skidder production message.

Skidder instruction

The object instruction for skidders (soi) is almost identical with the forwarder instruction with the exception that element ProductUserID is replaced with element SpeciesGroupUserID as illustrated below:



Observe that the skidder also needs an spi-file in order to include the necessary information regarding the relevant SpeciesGroup.

Yarder production

The basic idea is to report each yarder-load with information about species and number of stems. This means that the Deliveries and Products are irrelevant when comparing with the forwarder.

The following existing elements and attributes need new enumeration values:

- Add “ypr” to attribute `messageCategory`
- Add “Yarder” to attribute `machineCategory`
- Add “Number of stems” to attribute `loadNumberOfCategory`

When comparing with the present standard for forwarder production the following new parameters are deemed to be required:

- **Carriage**
 - Brand/Model/Year
- **Extraction Method**
 - Grapple
 - Choker/Winch
- **Stem Presentation**
 - Un-bunched – stems are generally not grouped into a bunch and are typically felled by a person with a chainsaw
 - Part Bunched – stems are roughly grouped into a bunch and are typically mechanically felled using a Dangle Head feller buncher
 - Bunched – stems are precisely grouped into a bunch and are typically mechanically felled using a Fixed Head feller buncher
 - Machine Fed – stems are placed or lifted by a separate machine into the grapple or for the grapple to grab
- **Tail Hold**

- Manual – The yarders wire ropes on the "backline" are shifted manually by people
- Mobile – The yarders wire ropes on the "backline" are shifted via a machine (excavator/bulldozer) that is driven by a person i.e. it's not manned all the time.
- Mobile Tele-Operated – The yarders wire ropes on the "backline" are shifted via a machine (excavator/bulldozer) that has been fitted with a remote control system and camera's so that the Yarder operator can remotely start the excavator up and drive it to the next Yarder line while never having to leave the Yarder
- **Rigging System** (how the wire rope systems are operated)
 - Northbend
 - Southbend
 - Shotgun
 - Highlead
 - Grabinski
 - Slack pulling carriage

It is suggested to add Carriage information to the Machine and to add ExtractionMethod , StemPresentation, RiggingSystem and TailHold to the Load as illustrated below. In most cases these values are not changed within an Object but they sometimes are depending e.g. on the terrain variability. This information is set by the operator.

Observe that the figure below is a simplification of the schema. The idea is to keep all elements and attributes from the fpr-schema except for those explicitly mention in previous section.

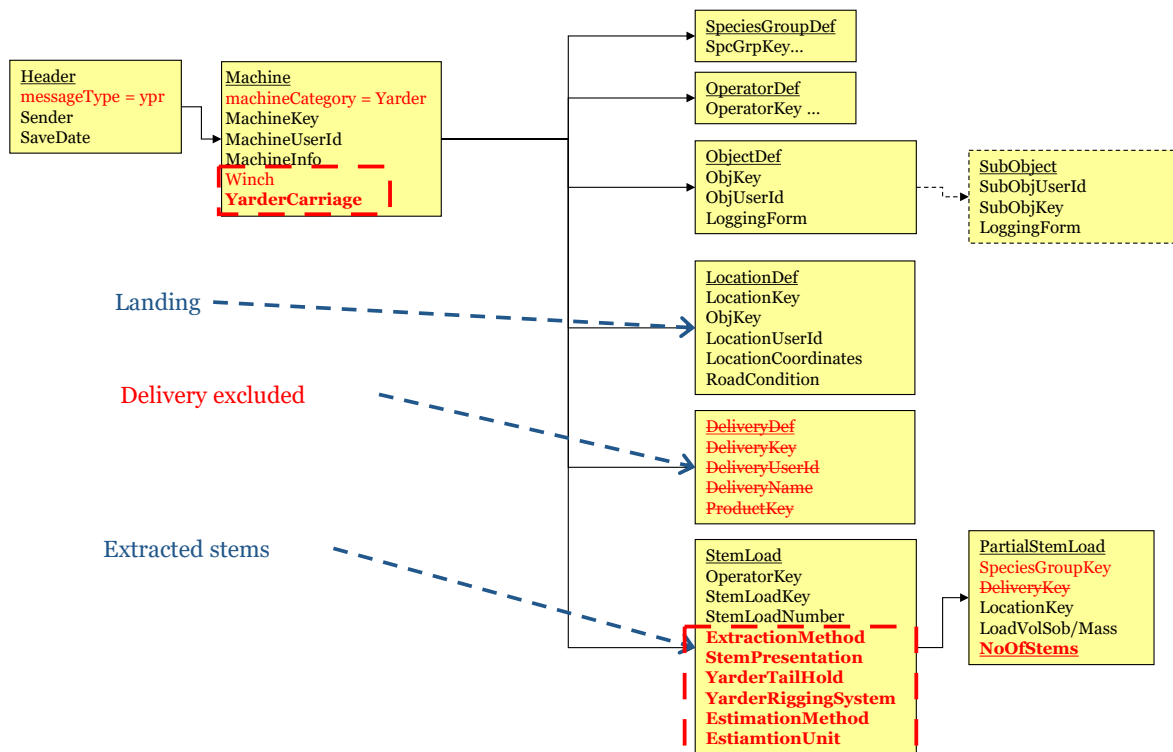
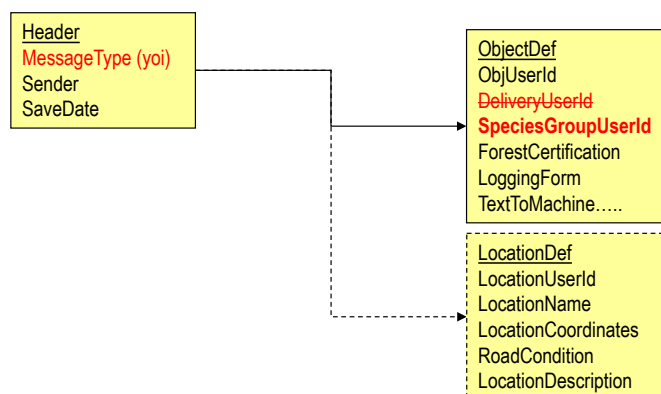


Figure 3. New or updated elements or attribute in red color. Observe that DeliveryDefinition is to be excluded.

Observe that machineCategory will not include the enumeration value "Yarder" until major version 4.0. Instead the value "Other" has to be used in the yarder production message.

Yarder instruction

The object instruction for yarders (yoi) is almost identical with the forwarder instruction with the exception that element ProductUserID is replaced with element SpeciesGroupUserID as illustrated below:



Observe that the yarder also needs an spi-file in order to include the necessary information regarding the relevant SpeciesGroup.

Detailed description of new parameters

EXTRACTION METHOD

Observe that a skidder can be equipped with both winch and grapple and therefore might use both of them even though using a winch is becoming less frequent.

Grapple

Skidder or yarder using a grapple to extract logs.





Choker

Skidder or yarder using a choker and/or winch to extract logs.



STEM PRESENTATION

Type of stem presentation which describes how stems are positioned before being picked up by skidder or yarder.

Un-bunched

Stems are generally not grouped into a bunch and are typically felled by a person with a chainsaw.



Part Bunched

Stems are roughly grouped into a bunch and are typically mechanically felled using a Dangle Head feller buncher.



Bunched

Stems are precisely grouped into a bunch and are typically mechanically felled using a Fixed Head feller buncher.



Machine Fed

Stems are placed or lifted by a separate machine into the grapple or for the grapple to grab.



TAIL HOLD

Type of tail hold describes how the wire ropes on the back line are moved from one extraction line to another.

Manual

The wire ropes on the backline are shifted manually by people across to the next Yarder extraction line. Slow and time consuming compared to the other 2 methods.

Mobile

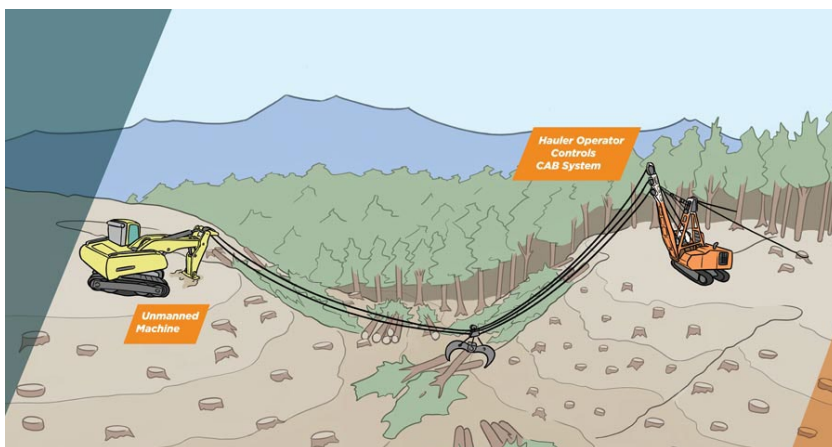
The wire ropes on the back line are shifted by machine across to the next Yarder extraction line. This machine is normally an excavator or bulldozer that gets driven by a person i.e. it's not manned all the time and a person usually will drive from the Yarder out to where the Tail

Hold machine is located, then shift the excavator/bulldozer across to the next Yarder line, then drive back to the Yarder.



Mobile Tele Operated

Similar to above, but the machine has been fitted with a remote control system and camera's so that the Yarder operator can remotely start the excavator up and drive it to the next Yarder extraction line while never having to leave the Yarders cab (this method is typically much quicker and doesn't need to take someone away from doing their normal job).



Rigging System

This parameter describes how the wire rope systems are operated. No detailed description of different rigging systems in this document.

Northbend

Southbend

Shotgun

Highlead

Grabinski

Slack pulling carriage

StanForD2010 Version 4.0

Schema re-structuring of ogi, mom and fpr

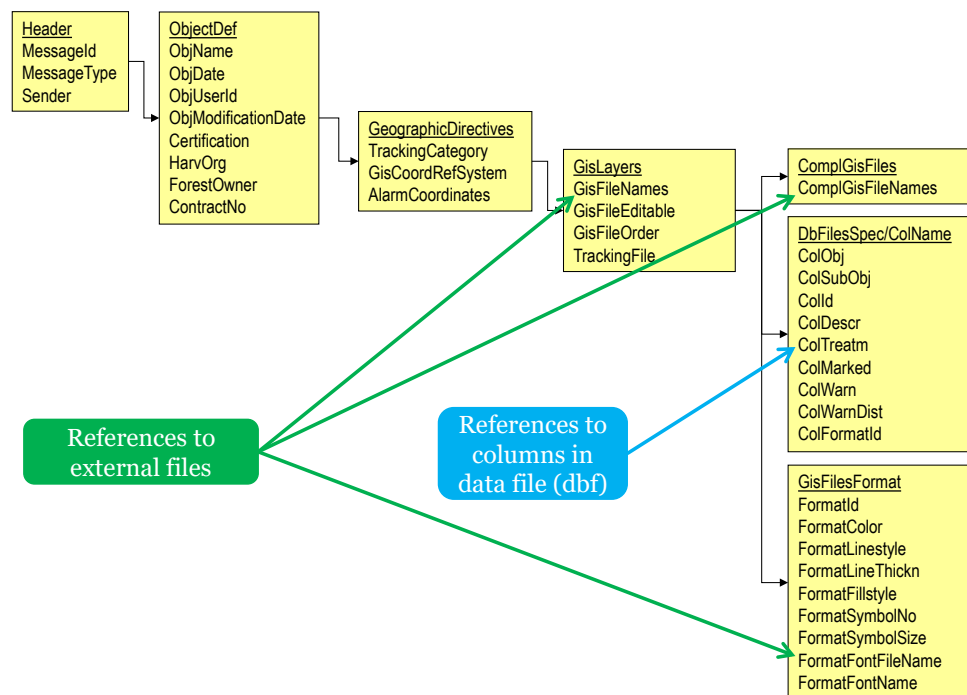
Contents

Contents.....	1
Schema re-structuring.....	2
ogi-file	2
mom-file	5
fpr-file	8
Implementation issues related to 4.0.....	9
Priorities (mandatory vs optional).....	10

Schema re-structuring

ogi-file

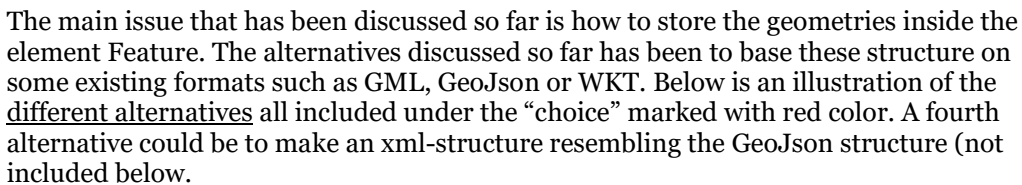
The possibility of re-modelling ogi has been discussed in order to make it simpler and not dependent on a number of other file types such as shp- and font-files as illustrated below. Another important driving force is that the present standard version, partly due to the complex structure, has been implanted in a heterogenous way.



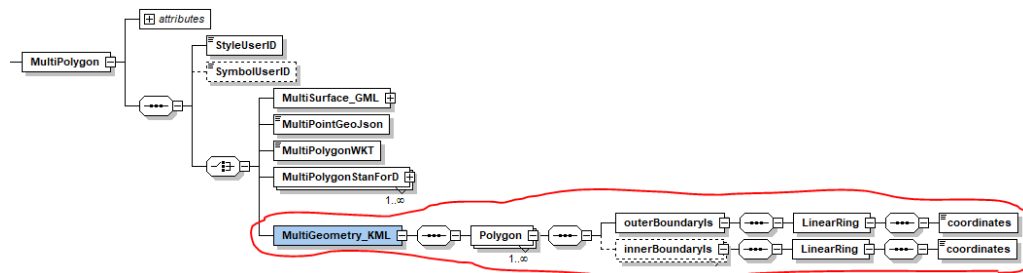
Other driving forces are recent proposals to:

- Implement a standardized set of symbols using formats such as svg and png.
- Use the format for simple and frequent (daily) updates of areas not to be harvested, e.g. protecting koala habitats ("sleeping areas" that can change daily)
- Open possibility to include text-based information aimed at the operator using html or url rather than today a complex pdf.

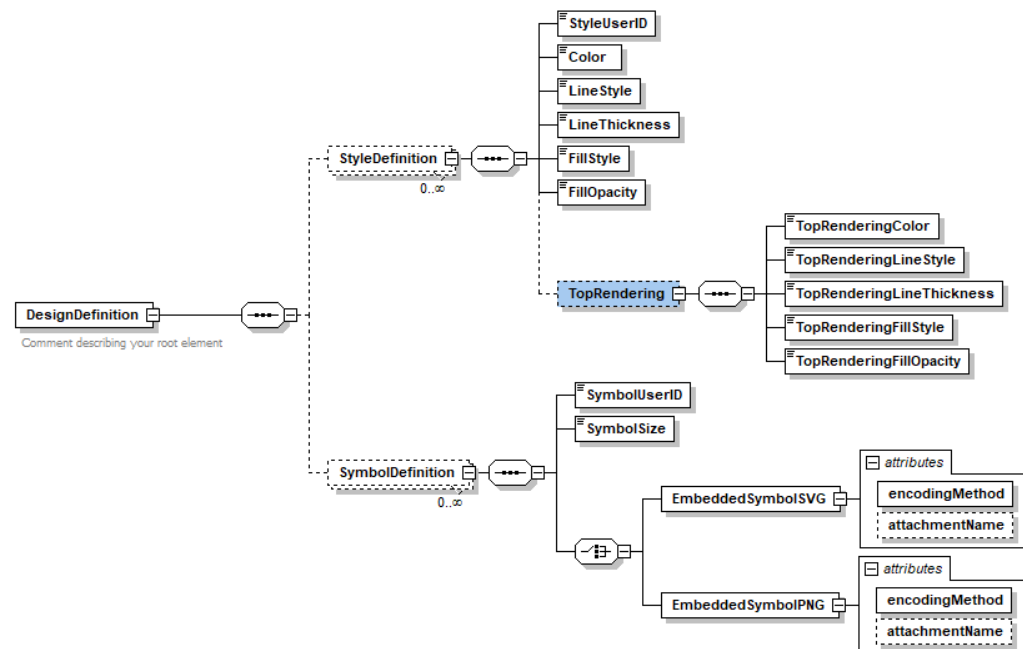
The new suggested structure means replacing the old element GeographicDirectives with a few new elements as illustrated below (observe that a number of ObjectDefinition-elements are not included):



Another alternative could be to include a structure inspired by KML as illustrated below.



Another fundamental change that is suggested is to move to formatting (designing) to a separate Design message and to utilize StyleUserID and SymbolUserID as a reference between the structure above and the Design message. The Design structure is illustrated below. Observe that a new parameter FillOpacity has been added when comparing with the old standard. Symbols (svg or png) can be included inside the message using element EmbeddedSymbolSVG/PNG using encodingMethod Base64.



The following figure illustrates how individual features are connected to specific symbol or style in the DesignDefinition.

Element	Feature 1	Feature 2	Feature 3
Name (label)	Main landing	Cutting	Route 66
Treatment	Both sides of road	Clear fell	
Description		Pine forest	
GeometryType	Point	Polygon	Line
Geometry	(30 10)	((30 10, 40 40, 20 40, 10 20, 30 10))	(30 10, 10 30, 40 40)
Order	1	2	3
DesignCategoryUserID		CuttingArea	RoadLine
SymbolUserID	Landing		Road
PresentationGroupUserID	Logging	Logging	Roads

DesignCategory-UserID	Line-Color	Line-Style	Line-Thickness
CuttingArea	Blue	Solid	3
StripRoadLine	Brown	Dot	1
RoadLine	Black	Solid	2
PowerLine	Red	Dash	1

SymbolUserID	Symbol	Symbol-Size
NoLogging		5
Landing		5
Road		4

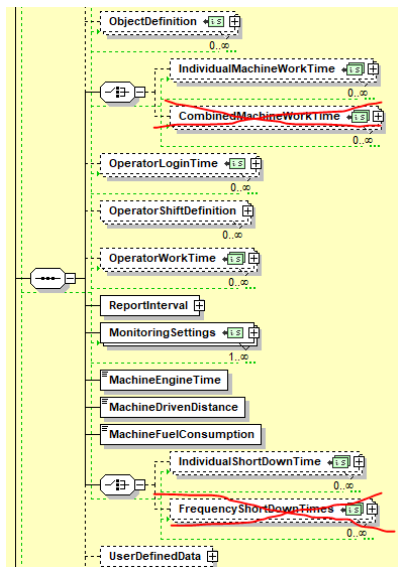
The intention is that Skogforsk will develop a default Design definition that can be used by any actor.

mom-file

Combined times excluded

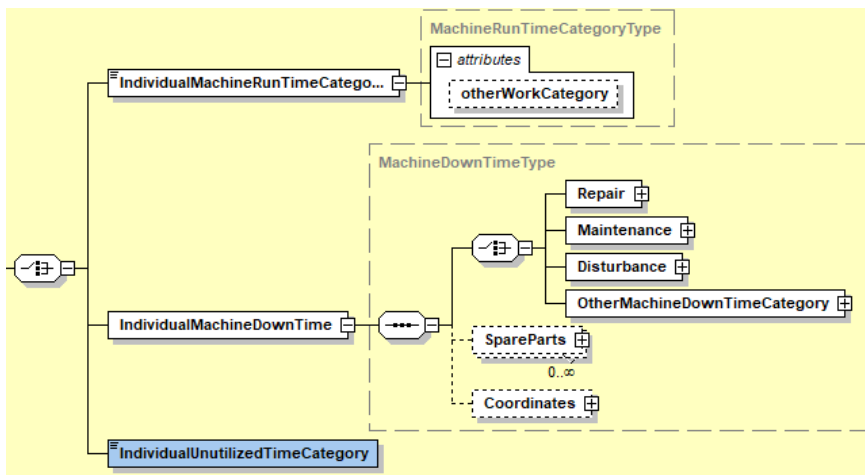
Excluding “combined” times from the mom schema would make the standard more consistent and simpler to use.

It is thus suggested that CombinedWorkTime and FrequencyShortDownTimes are excluded from the mom schema:

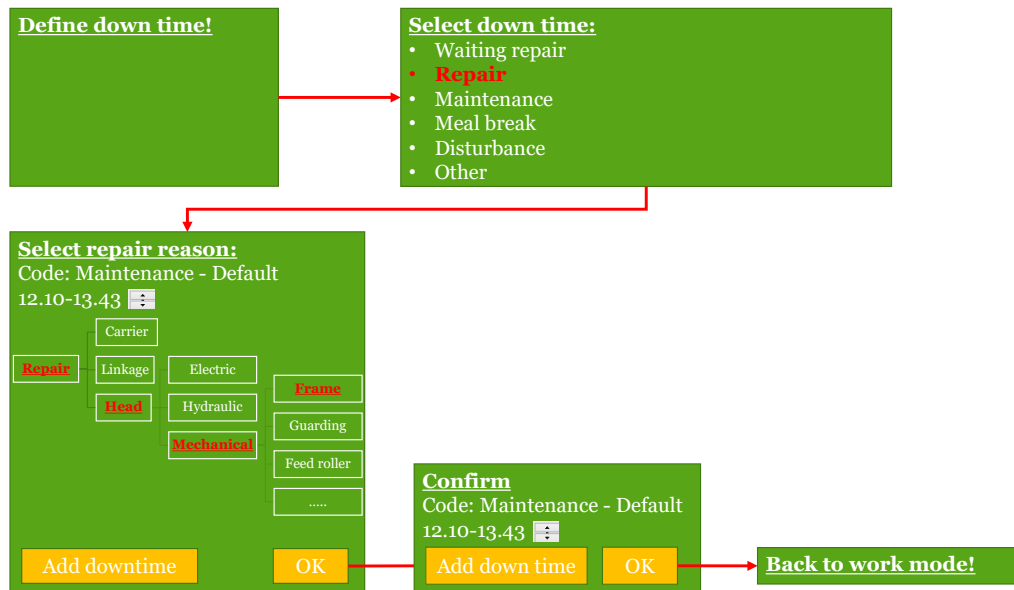


Simplified down time reasons

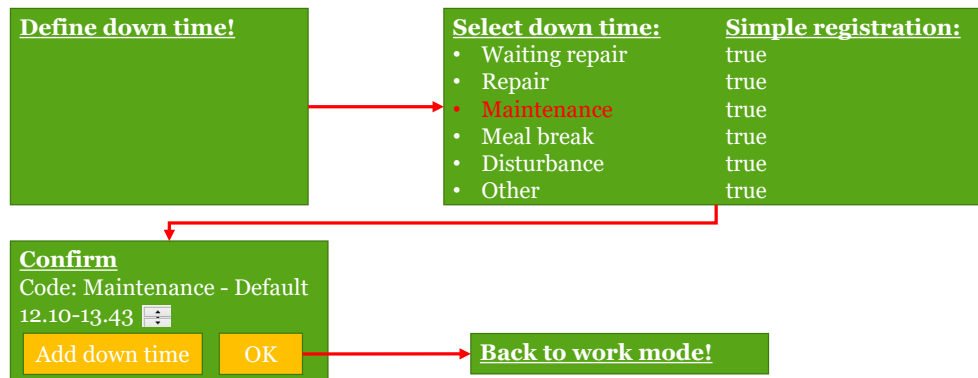
Another improvement could be to make it possible to simplify registration of down time reasons. Basically not requiring the operator to go into any more details than illustrated below where “No reason” could be used for “Shortened registration”, “Repair”, “Maintenance”, “Disturbance” and “Other Machine down time”:



Below is a flow chart illustrating how a down time is normally registered today. The operator must navigate through a large number of steps in order to select e.g. a specific repair reason.

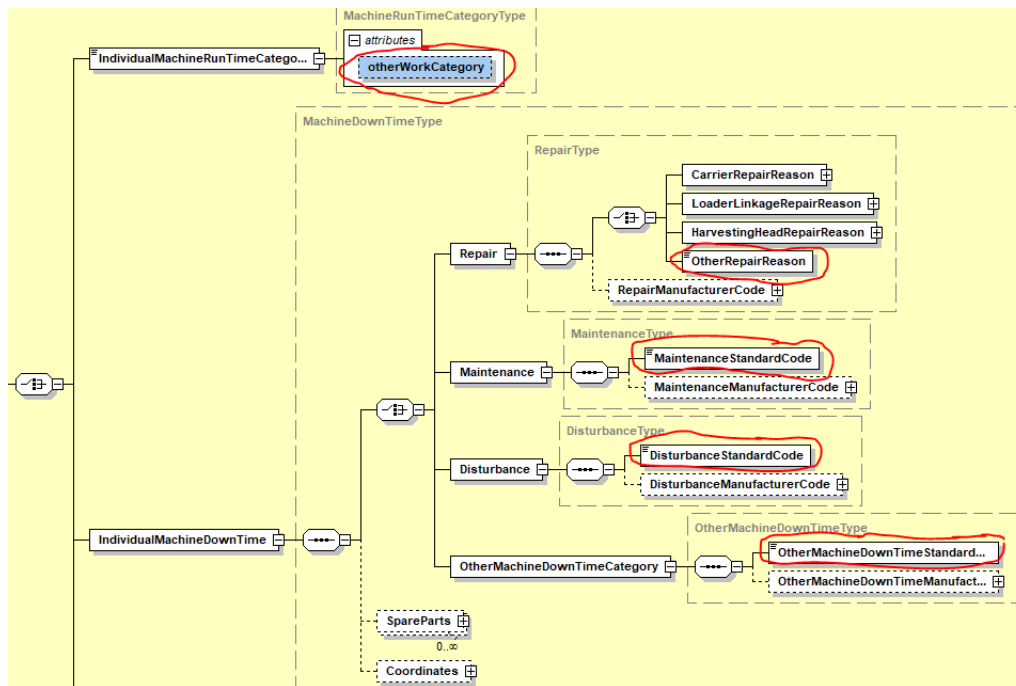


Using a simplified interface it would be possible to register a down time reason much quicker:



Below is an illustration of a simplified interface where no detailed down time codes are mandatory (based on the present Timbermatic interface):

It is suggested that a new enumeration value "Default" is added to the following elements in order to be able to communicate that a simplified registration of down times has been used: OtherWorkCategory, OtherRepairReason, MaintenanceStandardCode, DisturbanceStandardCode and OtherMachineDownTimeStandardCode



Move “Unproductive work”

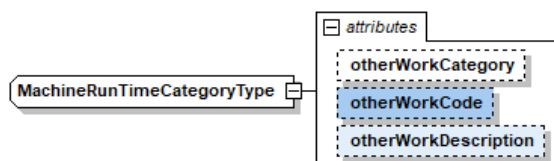
There is a down time enumeration for “Unproductive work”. This should probably be registered as “Other work time” since the machine is probably moving.

Extra code for other work

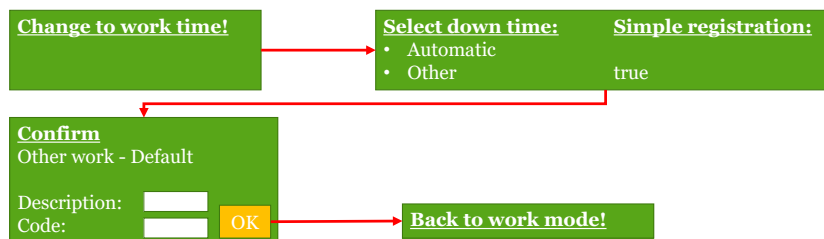
Extra code for other work times

- Make it possible for operator to register any non-standardized code for other work times.
- Enumeration list not enough, e.g. SCA is using the following codes (today handled by specifying separate “artificial” operators):
 - Other work 950
 - Road travel 952
 - Forest roads 954
 - Special driving 958
 - Build road on wet land (Kavling) 961
 - Build bridge over troubled water 964

It is suggested that we add two new attributes set manually by the operator (otherWorkCode and otherWorkDescription):



Example illustrating user interface:



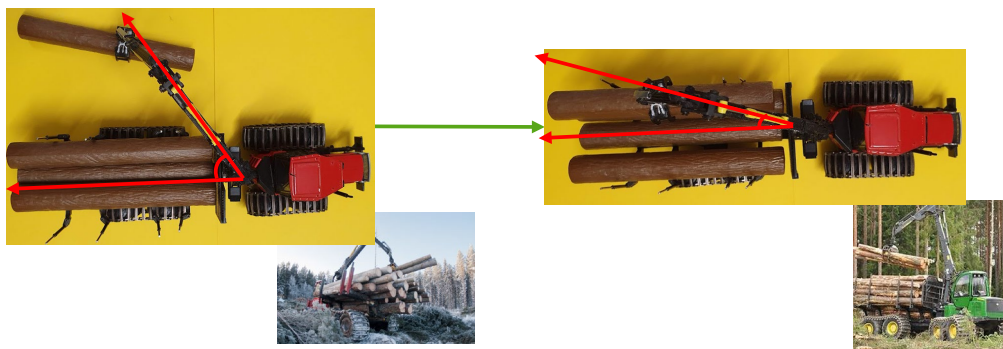
Comment implementation of 4.0 mom

- Non need to support both 4.0 and 3.x in machine control systems
- It is ok to support only 4.0 since the new standard version probably affects the GUI!

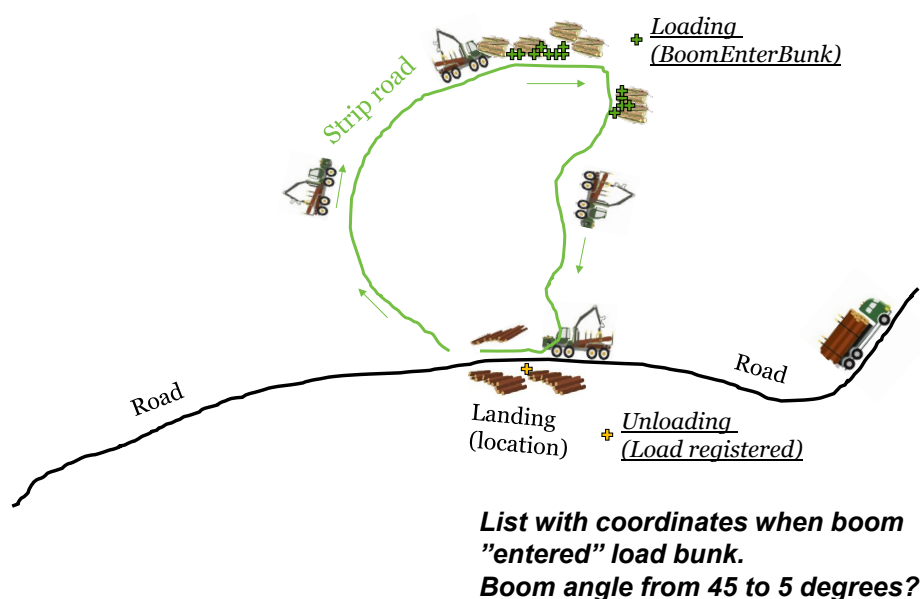
fpr-file

Adding more detailed information related to where logs are loaded in the forest would make it possible to improve traceability (connect harvester data with forwarder data). The aim is to find a model that is not relying on time stamps or Tracking data.

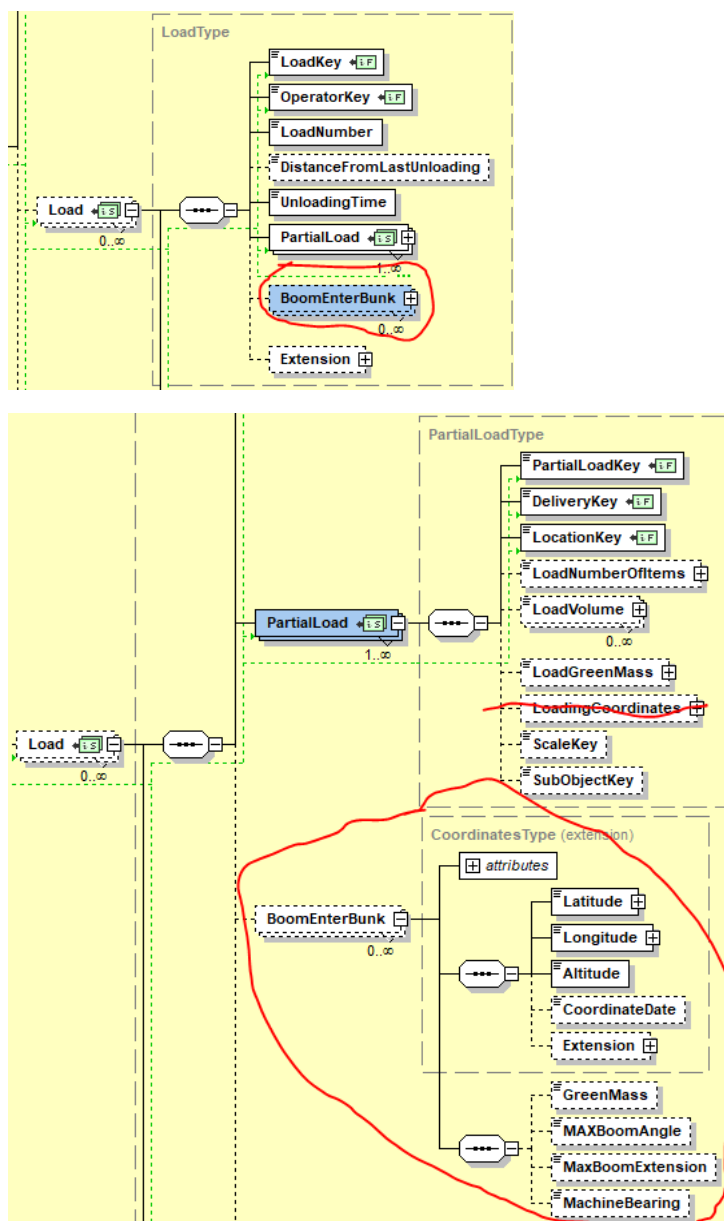
An idea could be to look at the possibility to register a “BoomEnterBunkCoordinates” each time the boom has been moved into the bunk (e.g. when boom angle changes from e.g. 45 degrees to 5 degrees).



These boom activities should be registered for each Load in order to make it possible to connect each “BoomEnterBunkCoordinates” to a specific load.



Below is a suggested structure for registering BoomEnterBunk



Implementation issues related to 4.0

Following issues/comments has been raised:

- Specify how to handle minor versions 3 and 4 parallel to each other (Antti Korkiakoski, JohnDeere). One suggestion from Henrik Persson (LogMax) was to state that control systems normally should run using one specific major version (e.g. do not mix 3.6 pin with 4.0 hpr).
- No significant changes to 3.6 should be made at the meeting in December 2020 (Jonas Thorsson, Dasa).

- No users are presently asking for a new major version (Heikki Komulainen, Ponsse).
- How do we avoid confusion in the “field” regarding the use of 3 vs 4? (Mikael Jakas, Logset).

Priorities (mandatory vs optional)

Henrik Persson, LogMax, 2019-06-28

New functional requirements could be added to the e.g.:

In order to publish ver 4 all applications must support the following functionalities even though these functions might not be used in all situations:

- *Tracking*
- *StemCode*
- *TopLogFreeBuck*
- *modificationRestricted*
- *UserDefinedData*
- *BoomPositioning*
- *StumpTreatment*

Adding these types of requirement could be difficult since the communication between controller and base machine might have limitations (e.g. boom positioning) or the need for some functions might not exist on certain markets at all (e.g. StumpTreatment or UserDefinedData).

One suggestion (2020-08-17) is to add attribute under Machine that indicates what functions are supported. Another solution could be to state that this is an issue that depends on the market requirements. Some major markets could probably “join forces” to give very clear signals about what is expected from 4.0 versions.

This is not a proposal but a possible starting point for a discussion!



Arlinger John

23 November 2020

MultiTreeProcessing and DistributionBucking

Anders Nylund, Komatsu, 2020-10-02

How to handle MTP-logs when working with distribution bucking? To be used in the algorithm or not?

If we have time the idea is to have a discussion regarding this issue.



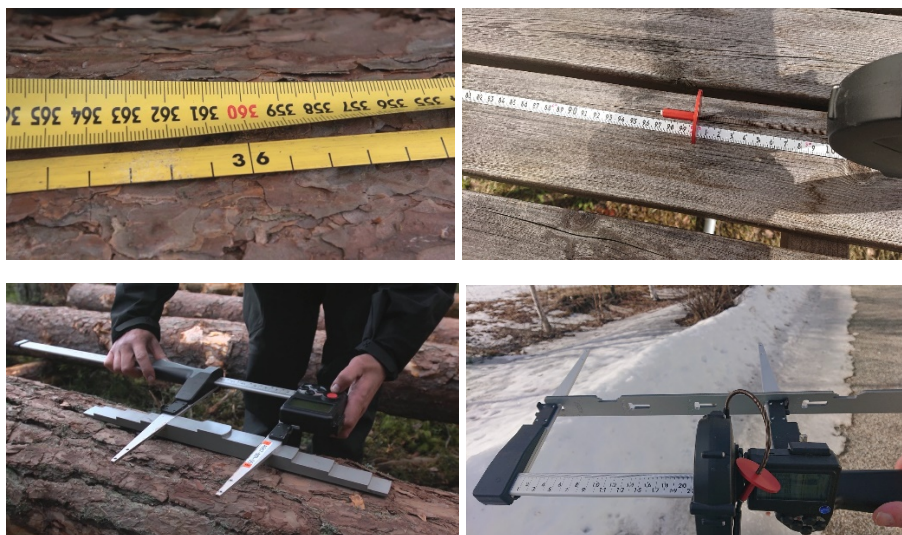
Arlinger John

24 November 2020

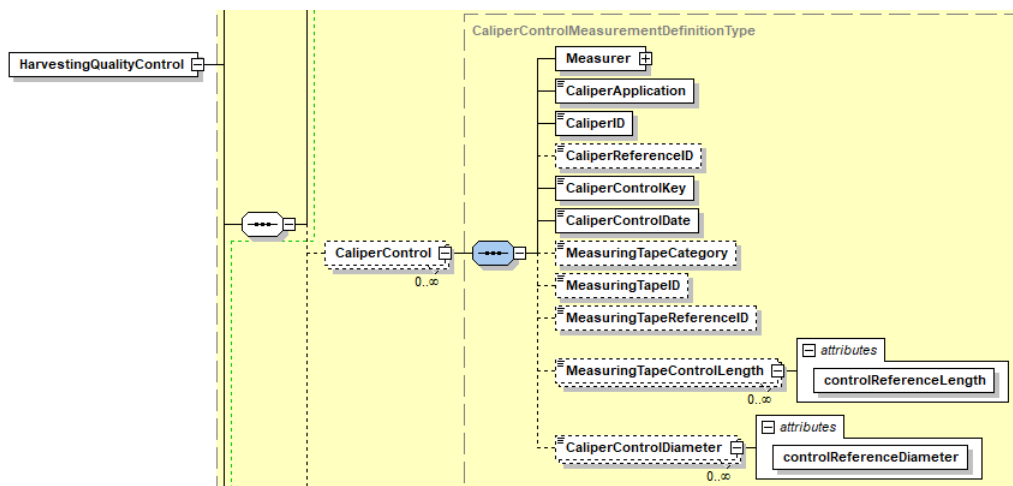
Caliper control test

Proposal based on discussion with Joakim Nygren, Haglöf, 2020-03-20

It is important to regularly check the handheld instruments used for controlling harvester measurements. The measuring tape is checked against a steel measuring tape, class 1. This also applies to digital measuring tapes, which must be calibrated if the length deviates from the template. The calliper is most easily checked with a caliper template.



Operators are supposed to control their caliper and measuring tape once a week. These controls are today logged in the calliper but not reported in a standardized way. It was therefore suggested that we add the following structure to the hqc-file.



The following new definitions were suggested to be added to the Hqc-schema:

Name	Unit	Data Type	Description
CaliperControl		nonNegative-Integer	Structure for control of caliper and measuring tape. Only registered when carrying out "Operator" measurement (M2)
Measurer		MeasurerType	Contact information of measurer
CaliperApplication	free text	String	Version of software used in caliper.
CaliperID	free text	String	Caliper identity information.
CaliperReferenceID	free text	String	Caliper reference identity information. Normally a fixed template with a number of fixed distances.
CaliperControlKey	sequential no	nonNegative-Integer	Unique caliper control identity, set by caliper. Must be totally unique for a certain Caliper.
CaliperControlDate	yyyy-mm-ddThh:mm:ss[+ -]hh:mm	StanForD2010-DateTimeType	Date and time of caliper control. Time stamp must have the following times registered: local time ±offset from UTC.
MeasuringTapeCategory	List	Enumeration list	Type of measuring tape: "Manual tape", "Digital tape".
MeasuringTapeID	free text	String	Measuring tape identity information.
MeasuringTapeReferenceID	free text	String	Reference measuring tape identity information. Normally a steel measuring tape class1.
MeasuringTapeControlLength	Cm	nonNegative-Decimal	Registered length when control measuring using measuring tape.
controlReferenceLength	Cm	nonNegative-Decimal	Reference length, length of the control object e.g. steel measuring tape class 1.
CaliperControlDiameter	Mm	nonNegative-Decimal	Registered diameter when control measuring using caliper.
controlReferenceDiameter	Mm	nonNegative-Decimal	Reference diameter, diameter of the control object e.g. caliper template.

It was decided (2020-05-07) to postpone this issue in order to study the possibility to create a separate message due to the fact that these controls are completely independent of control stem measuring as well as the harvester. It was also

suggested at the meeting to add elements for registering the identities of the caliper template and the measuring tape (included in table above).

The proposed structure has been implemented for testing by Haglöfs since there were some urgent needs for this information on the Swedish market. The structure is stored under an extension element as illustrated in the example below.

HarvestingQualityControlHeader						
Machine machineCategory=Harvester						
Extension						
CaliperControl (3)						
	Measurer	CaliperApplicati...	CaliperID	CaliperReferenc...	CaliperControlK...	CaliperControlDate
1	Measurer	SKALMAN7.20b	30057	10200A	2	2020-08-25T15:41:25.0+01:00
2	Measurer	SKALMAN7.20b	30057	10200A	3	2020-09-07T09:49:00.0+01:00
3	Measurer	SKALMAN7.20b	30057	10200A	4	2020-09-14T18:45:21.0+01:00

CaliperControlDate	MeasuringTape...	MeasuringTapeID	MeasuringTape...	MeasuringTapeControlLength	CaliperControlDiameter
2020-08-25T15:41:25.0+01:00	Digital tape	08201	20300B	MeasuringTapeControlLength (2)	CaliperControlDiameter (2)
				controlReferenceLength	controlReferenceDiameter
				1 100.0	1 10.0
				2 600.0	2 30.0
2020-09-07T09:49:00.0+01:00	Digital tape	08201	20300B	MeasuringTapeControlLength (1)	CaliperControlDiameter (2)
				controlReferenceLength	controlReferenceDiameter
				1 100.0	1 10.0
					2 30.0
2020-09-14T18:45:21.0+01:00	Manual tape	12345C	20300B	MeasuringTapeControlLength (2)	CaliperControlDiameter (2)
				controlReferenceLength	controlReferenceDiameter
				1 100.0	1 10.0
				2 500.0	2 30.0